

THREE-DIMENSIONAL ELASTO-STATIC PROBLEMS
USING ISOPARAMETRIC FINITE ELEMENTS

Dale Eugene Hanson

Gaylord
SHELF BINDER
Syracuse, N. Y.
Stockton, Calif.

United States Naval Postgraduate School



THESIS

THREE-DIMENSIONAL ELASTO-STATIC PROBLEMS
USING
ISOPARAMETRIC FINITE ELEMENTS

by

Dale Eugene Hanson

Thesis Advisor:

G. Cantin

June 1971

Approved for public release; distribution unlimited.

T139121

Three-Dimensional Elasto-Static Problems
Using
Isoparametric Finite Elements

by

Dale Eugene Hanson
Lieutenant, United States Navy
B.S., United States Naval Academy, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

The objective of the project described in this report was to develop a computer system which would generate the required input data for a structural analysis of three-dimensional elasto-static problems using isoparametric finite elements. Element connectivity, nodal point coordinates, consistent gravity loads, and consistent pressure loads are generated. A variety of algorithms are used in the system to reduce the amount of input data required. The computer system and a sample problem are discussed.

TABLE OF CONTENTS

I.	Introduction -----
II.	Isoparametric Finite Elements -----
A.	Displacement Functions -----
B.	Transformations -----
C.	Shape Functions -----
1.	Quadratic Elements -----
2.	Cubic Elements -----
III.	Discussion of Computer Programs -----
A.	Element Connectivity -----
B.	Solution Time and Space Requirements -----
C.	Nodes On Boundary Segments -----
D.	Corner Nodes -----
E.	Mid-side Nodes -----
F.	Consistent Gravity Load -----
G.	Consistent Pressure Load -----
H.	Plot of Structure Mesh -----
IV.	Input Data Preparation -----
A.	Variables -----
B.	Data Cards -----
1.	Element Field Data -----
2.	Problem Identification -----
3.	Nodal Point Coordinates -----
4.	Segment Divisions -----
5.	Load Data -----

6. Pressure Loading Data -----

7. Plot Title Cards -----

8. Stop Trap -----

V. Conclusions -----

VI. Recommendations -----

Appendix A Generating A Mesh with QUAMEG -----

Appendix B A Classical Problem Solved with TRISOP -----

Appendix C Computer Listing of QUAMEG -----

Appendix D Computer Listing of CUMEG -----

References -----

Initial Distribution List -----

Form DD 1473 -----

LIST OF TABLES

Table I. Listing of Input Data Cards -----

Table II. Displacement at the Applied Load for a
Pinched Cylinder -----

LIST OF FIGURES

- Figure 1. Quadratic Element -----
- Figure 2. Cubic Element -----
- Figure 3. Abstract Structure -----
- Figure 4. Boundary Segment -----
- Figure 5. Bounding Surface -----
- Figure 6. Pressurized Element -----
- Figure 7. Problem Structure -----
- Figure 8. Front Face -----
- Figure 9. Back Face -----
- Figure 10. Front Face of Structure -----
- Figure 11. Back Face of Structure -----
- Figure 12. Generated Mesh For Front Face -----
- Figure 13. Generated Mesh For First Intermediate Slice -----
- Figure 14. Generated Mesh For Second Intermediate Slice -----
- Figure 15. Generated Mesh For Third Intermediate Slice -----
- Figure 16. Generated Mesh For Back Face -----
- Figure 17. Generated Mesh For Front Face -----
- Figure 18. Pinched Cylinder -----

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Professor Gilles Cantin who introduced this author to the finite element method of structural analysis and to the concept of an automatic mesh generator and who provided his moral support, encouragement, and assistance throughout the development of this project.

I. INTRODUCTION

For a structural analysis of any non-trivial elasto-static problem by the finite element method, a large amount of data must be provided to the computer system which performs the calculations. The required information includes the arrangement of nodes within each element, the coordinates of all nodes, and the applied loads. After the numerical values of this data have been determined, which is a large problem in itself, the information must be supplied to the computer system. This is normally accomplished by punching the appropriate data on computer cards. The typing of these cards is also a difficult and time-consuming job. More important, however, is the ever-present possibility of human errors during the performance of these tasks. If an error is not detected prior to submitting the data for processing, expensive computer time would be wasted.

With the above considerations in mind, it is readily apparent that a computer system, which would generate most of the data required by the primary computer system for a complete structural analysis, would be very useful. In addition, this mesh generator should require as little input data as possible.

References 5, 8, and 10 contain reports of efforts which have been directed at coding computer systems which will aid the analyst in the interpretation of output data. A few computer systems, which are reported in Refs. 1, 2, and 9, have been coded to automatically generate input data. However, these input data generation systems

are either for types of finite elements other than isoparametric elements or for two-dimensional elements.

Two input data generator systems for three-dimensional isoparametric finite elements will be presented in this thesis. One system is for quadratic elements and the second system is for cubic elements.

II. ISOPARAMETRIC FINITE ELEMENTS

The isoparametric concept will be discussed in this chapter in sufficient detail to establish only the procedures for constructing isoparametric elements and the validity of relationships used to obtain the consistent gravity and pressure loads. The readers are referred to Refs. 4 and 7 for further details of the isoparametric concept.

A. DISPLACEMENT FUNCTIONS

The displacements of nodes in any one element are obtained by using shape functions which are defined in a system of dimensionless coordinates (ξ, η, ζ) that are unique for that element. The coordinates range from minus one to plus one.

$$\begin{aligned} u(x, y, z) &= \sum_i N_i(\xi, \eta, \zeta) u_i \\ v(x, y, z) &= \sum_i N_i(\xi, \eta, \zeta) v_i \\ w(x, y, z) &= \sum_i N_i(\xi, \eta, \zeta) w_i \end{aligned} \tag{1}$$

where $N_i(\xi, \eta, \zeta)$ are the shape functions and u_i, v_i, w_i are the displacement components of node "i" in a global reference system (x, y, z) , where "i" assumes values from 1 to the total number of nodes in the element. This global reference system is then related to the dimensionless coordinates by the same shape functions as follows:

$$\begin{aligned} x(\xi, \eta, \zeta) &= \sum_i N_i(\xi, \eta, \zeta) x_i \\ y(\xi, \eta, \zeta) &= \sum_i N_i(\xi, \eta, \zeta) y_i \\ z(\xi, \eta, \zeta) &= \sum_i N_i(\xi, \eta, \zeta) z_i \end{aligned} \tag{2}$$

where x_i , y_i , z_i are the coordinates of node "i" in the global reference system [4].

The shape functions referred to above have been obtained from [7]. They are listed in section 2.C.

B. TRANSFORMATIONS

In order to obtain the consistent gravity and pressure loads, the volumetric increment, $dx dy dz$, must be found in terms of the dimensionless coordinates. That operation requires the Jacobian matrix [4].

$$[J] = \begin{bmatrix} \partial x / \partial \xi & \partial y / \partial \xi & \partial z / \partial \xi \\ \partial x / \partial \eta & \partial y / \partial \eta & \partial z / \partial \eta \\ \partial x / \partial \zeta & \partial y / \partial \zeta & \partial z / \partial \zeta \end{bmatrix} \quad (3)$$

The element of volume is then transformed as follows:

$$dV = \det[J] d\xi d\eta d\zeta \quad (4)$$

C. SHAPE FUNCTIONS

In this listing of the shape functions, attention is directed to the tri-quadratic and tri-cubic elements, for which the mesh generator programs were written.

1. Quadratic Elements

See Figure 1 for identification of the nodes.

Corner nodes: 1, 3, 5, 7, 13, 15, 17, and 19.

$$N_i = (1/8)(1 + \xi_o)(1 + \eta_o)(1 + \zeta_o)(\xi_o + \eta_o + \zeta_o - 2) \quad (5)$$

where $\xi_o = \xi_i \xi$ and ξ_i is ± 1 ; similarly for η_o and ζ_o .

Mid-side nodes: 2, 6, 14, and 18.

$$N_i = (1/4)(1 - \xi^2)(1 + \eta_o)(1 + \zeta_o) \quad (6)$$

Mid-side nodes: 4, 8, 16, and 20.

$$N_i = (1/4)(1 - \eta^2)(1 + \xi_o)(1 + \zeta_o) \quad (7)$$

Mid-side nodes: 9, 10, 11, and 12.

$$N_i = (1/4)(1 - \zeta^2)(1 + \xi_o)(1 + \eta_o) \quad (8)$$

2. Cubic Elements

See Figure 2 for identification of nodes.

Corner nodes: 1, 4, 7, 10, 21, 24, 27, and 30.

$$N_i = (1/64)(1 + \xi_o)(1 + \eta_o)(1 + \zeta_o)[9(\xi^2 + \eta^2 + \zeta^2) - 19] \quad (9)$$

Mid-side nodes: 2, 3, 8, 9, 22, 23, 28, and 29.

$$N_i = (9/64)(1 - \xi^2)(1 + 9\xi_o)(1 + \eta_o)(1 + \zeta_o) \quad (10)$$

where $\xi_i = \pm 1/3$, $\eta_i = \pm 1$, and $\zeta_i = \pm 1$.

Mid-side nodes: 5, 6, 11, 12, 25, 26, 31, and 32.

$$N_i = (9/64)(1 - \eta^2)(1 + 9\eta_o)(1 + \xi_o)(1 + \zeta_o) \quad (11)$$

where $\eta_i = \pm 1/3$, $\xi_i = \pm 1$, $\zeta_i = \pm 1$.

Mid-side nodes: 13, 14, 15, 16, 17, 18, 19, and 20.

$$N_i = (9/64)(1 - \zeta^2)(1 + 9\zeta_o)(1 + \xi_o)(1 + \eta_o) \quad (12)$$

where $\zeta_i = \pm 1/3$, $\xi_i = \pm 1$, and $\eta_i = \pm 1$.

Note that the convention for numbering of nodes within any element is to begin at the location where $\xi = \eta = \zeta = 1$ and to consecutively number the nodes, proceeding in a counter-clockwise direction around the ζ -axis (see Figures 1 and 2).

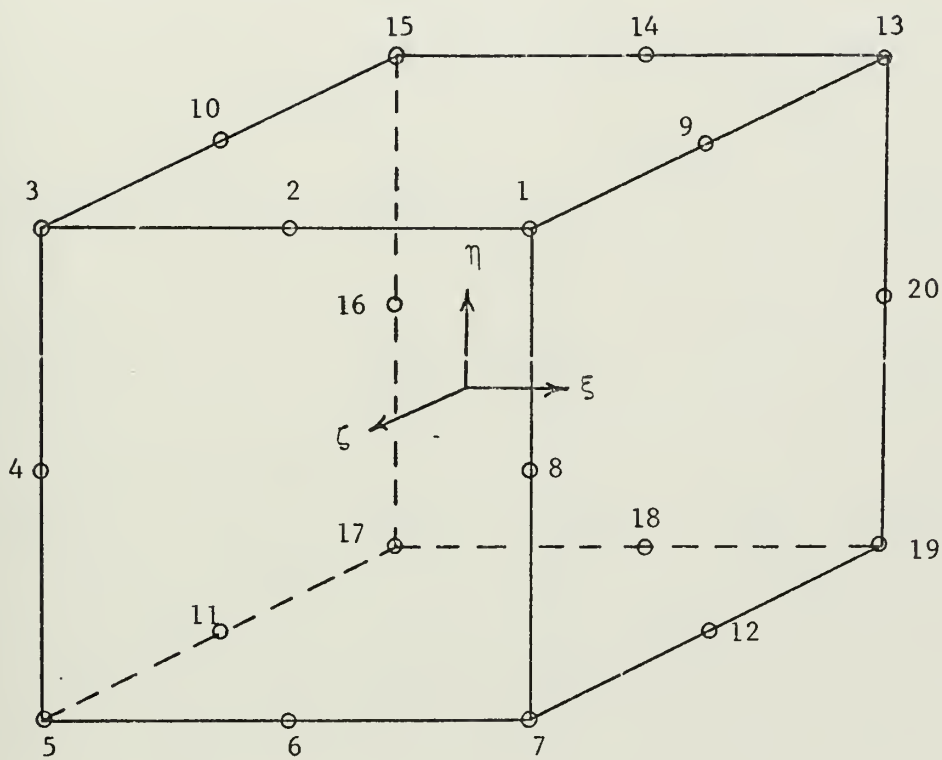


Figure 1. Quadratic Element

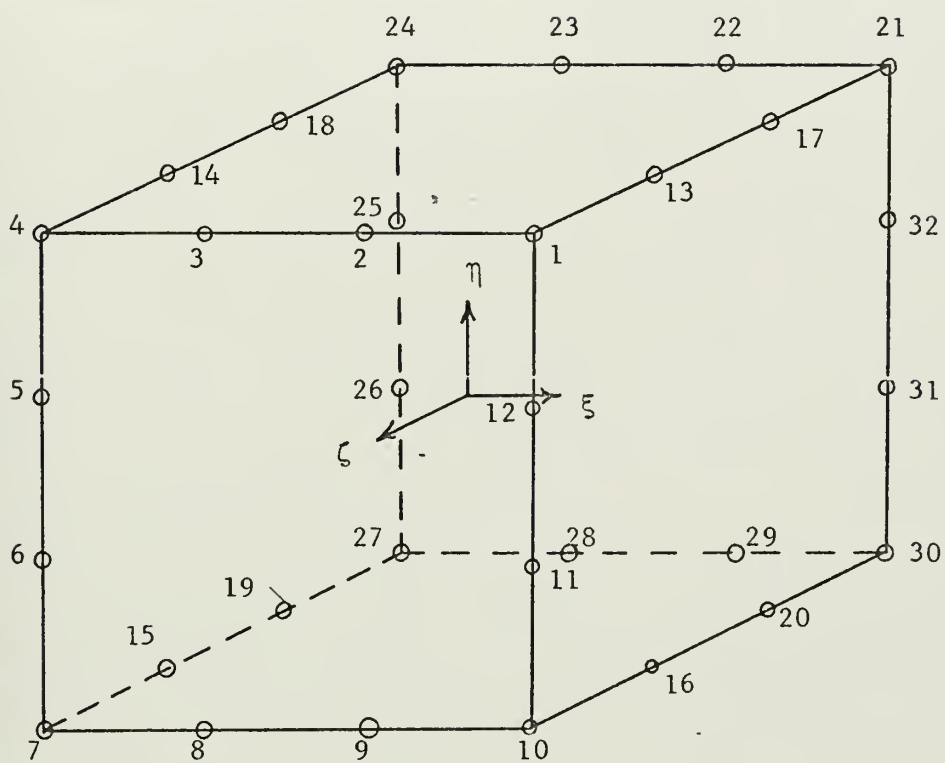


Figure 2. Cubic Element

III. DISCUSSION OF COMPUTER PROGRAMS

The computer programs presented in this thesis were written to support a computer system called TRISOP. TRISOP was coded by Professor G. Cantin of the U. S. Naval Postgraduate School, Monterey, California. TRISOP performs a structural analysis of three-dimensional elastostatic problems using isoparametric finite elements. Information required by TRISOP for any problem includes general mesh parameters, title cards, format statements, and material properties; Young's modulus of elasticity, Poisson's ratio, and the coefficient of thermal expansion. Fifteen cards are needed to supply this data to TRISOP. The bulk of the data required by TRISOP is the element connectivity, structure geometry, loading, and boundary conditions. The boundary conditions will vary with each problem; therefore, no effort has been made to automate this process.

The mesh generating computer programs discussed in this chapter compute the remainder of the data required by TRISOP: element connectivity, coordinates of nodal points, consistent gravity load vector, and consistent pressure load matrix. Additional information which is helpful in preparing an input deck of cards for TRISOP is also computed. One of these programs generates data for quadratic elements and the second program generates data for cubic elements. These programs will be called QUAMEG and CUMEG.

A. ELEMENT CONNECTIVITY

A structure of any shape can be visualized as a cubical piece of plastic material which can be stretched, compressed, and otherwise

appropriately deformed in such a way that it will assume the geometry of the structure. The element connectivity is determined when the structure is in this initial abstract cubical shape. The dimensionless coordinates of this abstract structure are ξ' , η' , ζ' .

Numbering of nodes is begun in a face which will have the fewest nodes. In that face, numbering proceeds sequentially along the edge having the smallest number of nodes as shown in Figure 3. This convention reduces the size of the half-bandwidth of the system, which directly affects the space and time required by the computer to solve the problem. The coordinate system is positioned such that node one is at $\xi' = 0$, $\eta' = 1$, and $\zeta' = 1$. An example is discussed in Appendix A.

For this mesh, the connectivities of the elements follow simple but tedious rules that were coded directly.

B. SOLUTION TIME AND SPACE REQUIREMENTS

The time required to solve the system of equations in the problem and the space required by the computer to store all data are important information. They must be determined before the problem is submitted for processing. The calculation of this information requires no additional input data and it is therefore included in QUAMEG and CUMEG to reduce the effort required of the programmer.

The time required to solve the equations is computed using an empirical formula which was constructed by Cantin [4]. This calculation has been included in the coding of QUAMEG and CUMEG.

The space required to store the various bits of information for a problem depends upon the total number of records and the

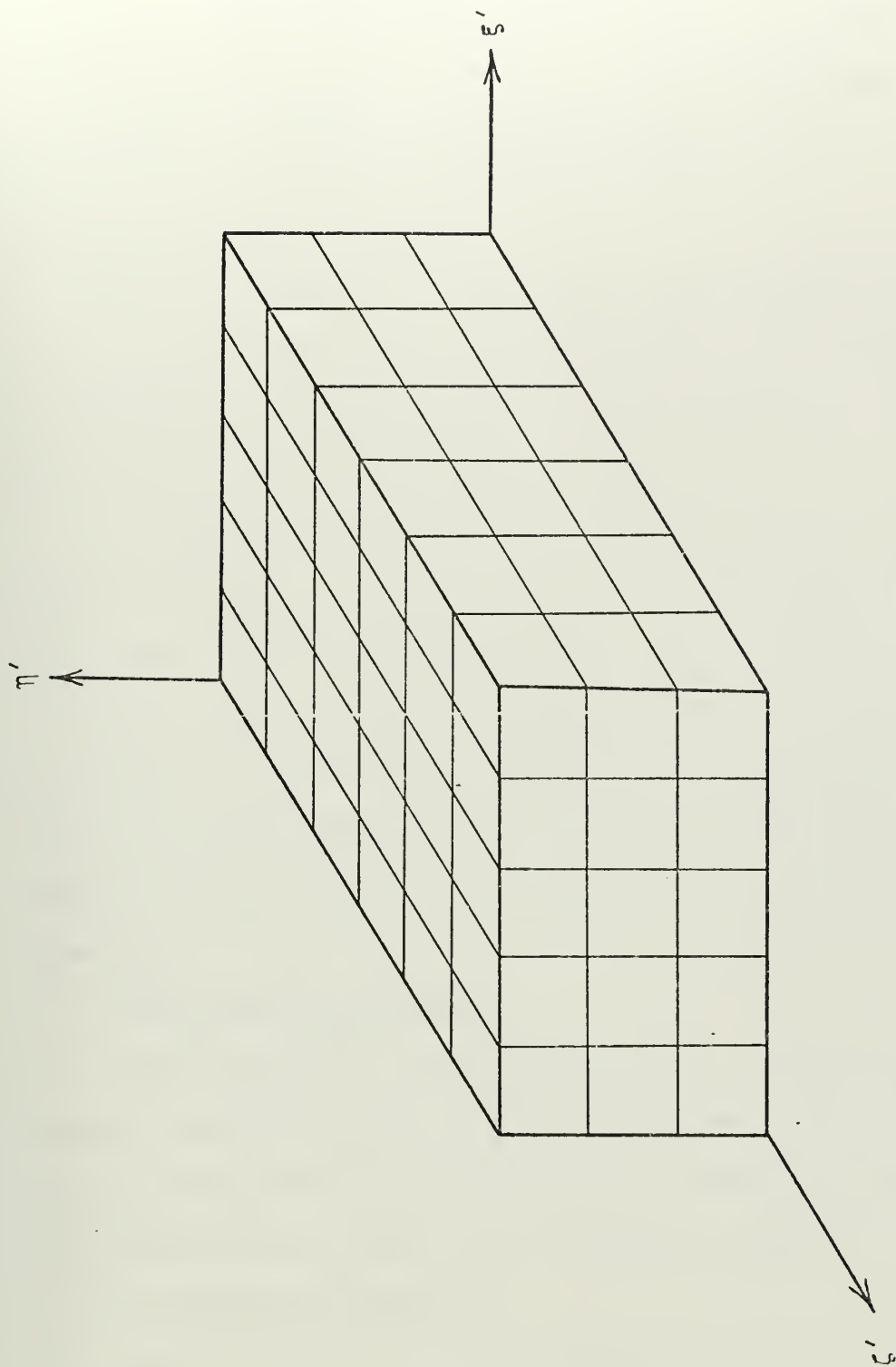


Figure 3. Abstract Structure

length of each record. These will vary with different types of computers. Simple formulas which generate the correct results for the IBM 360/67 computer used at the U. S. Naval Postgraduate School are employed in the coding of QUAMEG and CUMEG.

C. NODES ON BOUNDARY SEGMENTS

Before the coordinates of nodes which are within the boundaries of a structure can be computed, the coordinates of all corner nodes on the edges of the bounding surfaces must be specified. This problem is simplified by the method discussed below.

The term "boundary segment" must first be defined for the purpose of this discussion. A segment must satisfy a few simple conditions which depend upon the orientation of the proposed segment in the structure and upon the type of coordinate system used to describe the geometry of the structure. QUAMEG and CUMEG are coded to allow the use of either rectangular or cylindrical coordinate systems. A combination of the two coordinate systems in any one problem is not permitted.

1. Rectangular Coordinates

When rectangular coordinates are used, a segment is any straight line between two corner nodes of the mesh. The element divisions along this line must be uniform. There are no other restrictions imposed upon a segment when rectangular coordinates are used.

2. Cylindrical Coordinates

When cylindrical coordinates are used, the radius and reference angle are taken in planes parallel to the x-y plane. In this case a segment can be an arc of circle centered on z and in a plane

parallel to plane x-y, or a straight line in a plane containing the z-axis. Again, element divisions along these lines must be uniform.

In order to determine the coordinates of nodes along a boundary segment, the coordinates of the corner nodes at the ends of the segment must be provided in the input data. In addition, the location of the segment within the mesh must be specified. The method of specifying the location of a segment is described in sections 4.B.3 and 4.B.4.

In Figure 4, the vector AB is easily obtained from the coordinates of nodes A and B. After calculating the values of increments of distances along x, y, and z, the intermediate nodes are easily obtained.

If the bounding surfaces of a structure are of an irregular configuration such that portions of the edges of that surface cannot be specified by segments, the coordinates of all nodes along this irregular portion of the bounding surface must be provided in the input data.

D. CORNER NODES

Once the coordinates of the nodes on all of the boundary segments are established, the coordinates of corner nodes on the bounding surfaces and on interior slices are calculated using a method described below. The bounding surfaces are considered in the following order (Figure 3): face 1 ($\zeta'=1$), face 2 ($\zeta'=0$), face 3 ($\xi'=0$), face 4 ($\xi'=1$), face 5 ($\eta'=1$), face 6 ($\eta'=0$). Slices which are each of constant ζ' are then considered until the coordinates of all corner nodes are calculated.

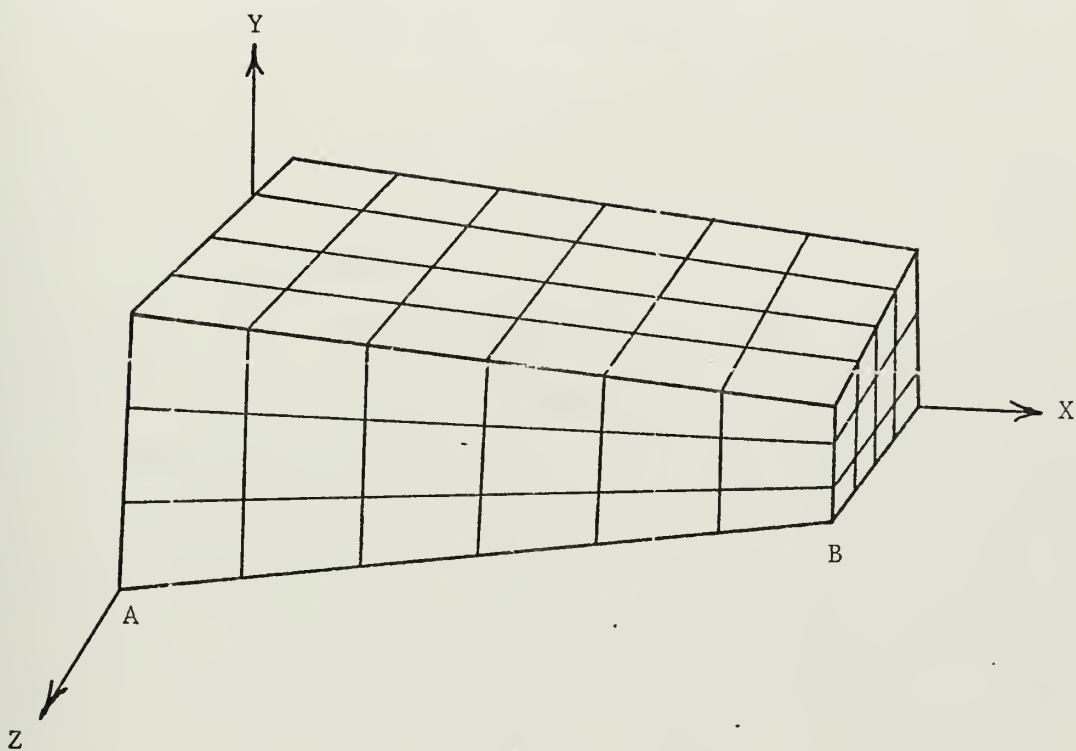


Figure 4. Boundary Segment

The method used for calculating the coordinates of corner nodes within the edges of a bounding surface or slice can best be understood with the example illustrated in Figure 5. In this example, the surface ABCD is the real surface for which a mesh is desired. The mesh is first constructed in the $\alpha\beta$ plane as shown.

At this time, the coordinates of all the corner nodes on the boundary are known. If the coordinates of the interior nodes are required to satisfy the Laplacian equation:

$$\begin{aligned}\nabla^2(x) &= 0 \\ \nabla^2(y) &= 0 \\ \nabla^2(z) &= 0\end{aligned}\tag{13}$$

where $\nabla^2 = \partial^2/\partial\alpha^2 + \partial^2/\partial\beta^2$

a set of acceptable interior nodes will be obtained. The dummy variables (α,β) are replaced by appropriate combinations of ξ' , η' , and ζ' , depending upon the location of the face being considered. For this computation, the standard Laplacian molecule of finite difference theory is used and the final coordinates are obtained by relaxation. Since the coordinates of the nodes within the edges of a face are arbitrary, the relaxation process can be stopped at any time, depending upon the degree of approximation desired. In this code, sixty iterations are used.

E. MID-SIDE NODES

After the coordinates of all corner nodes have been established, the coordinates of the mid-side nodes are computed by averaging the coordinates of adjacent corner nodes. Although the mid-side nodes need not be located exactly mid-way between the corner nodes, it is the simplest means of determining their coordinates.

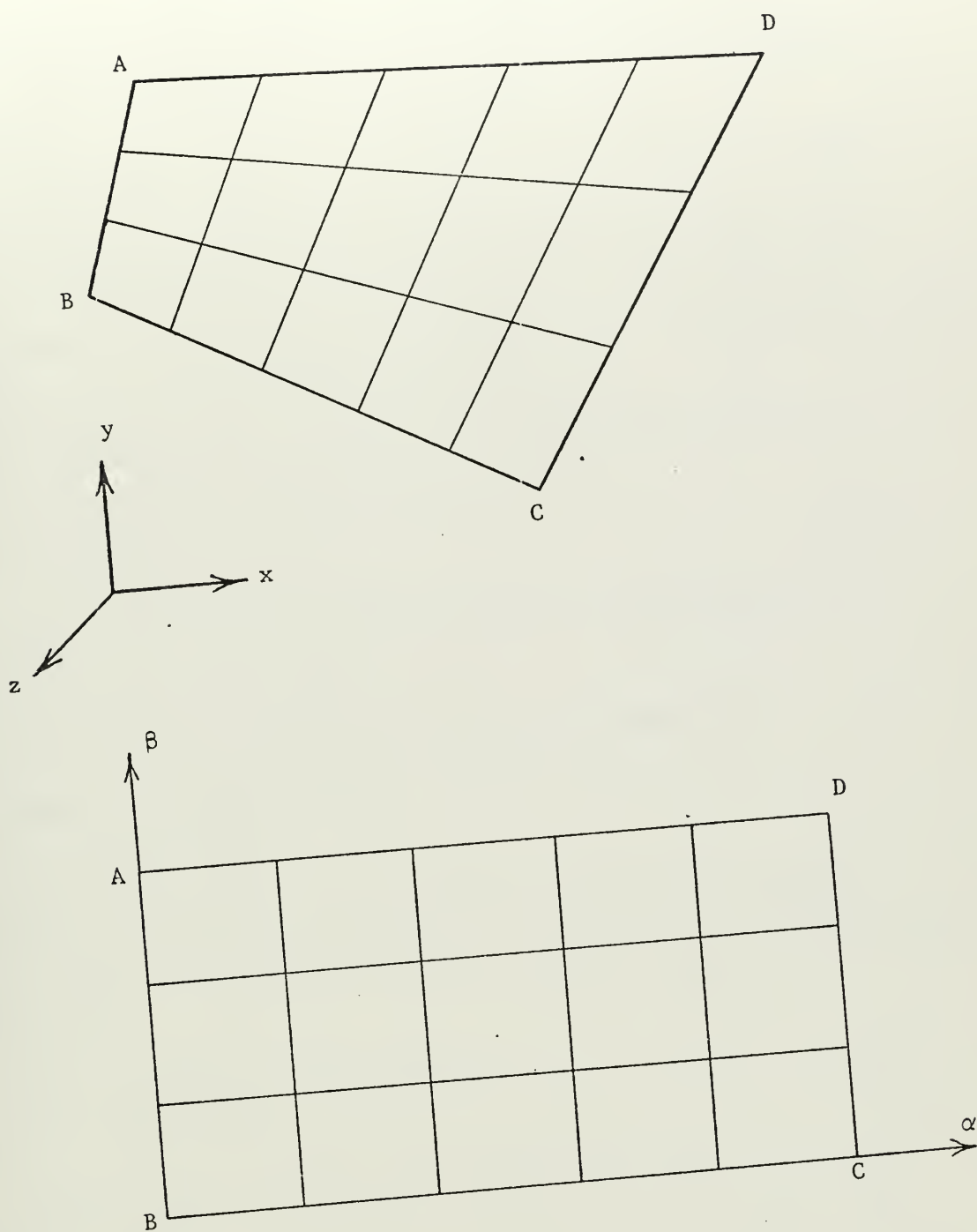


Figure 5. Bounding Surface

F. CONSISTENT GRAVITY LOAD VECTOR

The z -axis has been chosen as the line of action of the gravity load. The work done by the force due to gravitational acceleration on a body is

$$U = \int_V \rho g w dV \quad (14)$$

where U is the work done; ρ is the mass density of the body; g is gravitational acceleration; w is the displacement of the body in the z -direction; and dV is the element of volume.

The work done by concentrated forces acting at each node in the structure mesh is, in matrix notation:

$$U = \langle V \rangle \{w_i\} \quad (15)$$

where $\langle V \rangle$ is the consistent gravity load vector and $\{w_i\}$ are the respective node displacements.

From Chapter 2, recall the following two expressions:

$$w(x,y,z) = \sum_i N_i(\xi,\eta,\zeta) \{w_i\}$$

$$dV = \det[J] d\xi d\eta d\zeta.$$

When these expressions are substituted into equation 14 and the result identified to equation 15, the consistent gravity load vector is obtained.

$$\langle V \rangle = \rho g \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \det[J] \langle N_i \rangle d\xi d\eta d\zeta \quad (16)$$

In this form, equation 16 is ideally suited for solution by Gaussian cubature. In QUAMEG, four Gauss points are used in each of the three directions, for a total of 64 points within each element. Five Gauss points are used in CUMEG. The weight factors and coordinate values are obtained from Ref. 11. The Jacobian matrix is determined by using three function statements and the shape functions

are determined with two function statements in Subroutine GRAP (Appendix C). These function statements are permuted to obtain the partial derivatives of the shape functions and the values of the shape functions for all nodes in the mesh.

G. CONSISTENT PRESSURE LOAD MATRIX

The consistent pressure load matrix for a uniformly distributed pressure acting on any face of an element is determined in a manner similar to that used for the gravity load.

The work done by the uniformly distributed pressure is:

$$U = \int_A (-p\bar{n} \, dA) \cdot (\bar{u}) \quad (17)$$

where \bar{n} is a unit normal vector (positive outward) and \bar{u} is the displacement vector at the surface (see Figure 6). The displacement vector \bar{u} can also be written as:

$$\bar{u} = u(x,y,z)\bar{i} + v(x,y,z)\bar{j} + w(x,y,z)\bar{k} \quad (18)$$

The vector $\bar{n}dA$ is found from differential geometry:

$$\bar{n}dA = (\partial\bar{r}/\partial\alpha) \times (\partial\bar{r}/\partial\beta) \, d\alpha d\beta \quad (19)$$

where $\bar{r} = x\bar{i} + y\bar{j} + z\bar{k}$ is a position vector of a point on the pressurized surface and α and β are dummy variables, standing for either ξ , η , or ζ .

The terms $\partial\bar{r}/\partial\alpha$ and $\partial\bar{r}/\partial\beta$ are simply two rows of the Jacobian matrix evaluated on the surface. After making appropriate substitutions and performing a few matrix manipulations, the expression reduces to:

$$U = \int_{-1}^1 \int_{-1}^1 \langle f(\alpha,\beta) \rangle \left\{ u_i^* \right\} \, d\alpha d\beta \quad (20)$$

where $\left\{ u_i^* \right\}$ is the vector of nodal coordinates for the eight nodes on the face and $f(\alpha,\beta)$ is a row vector of functions of α and β .

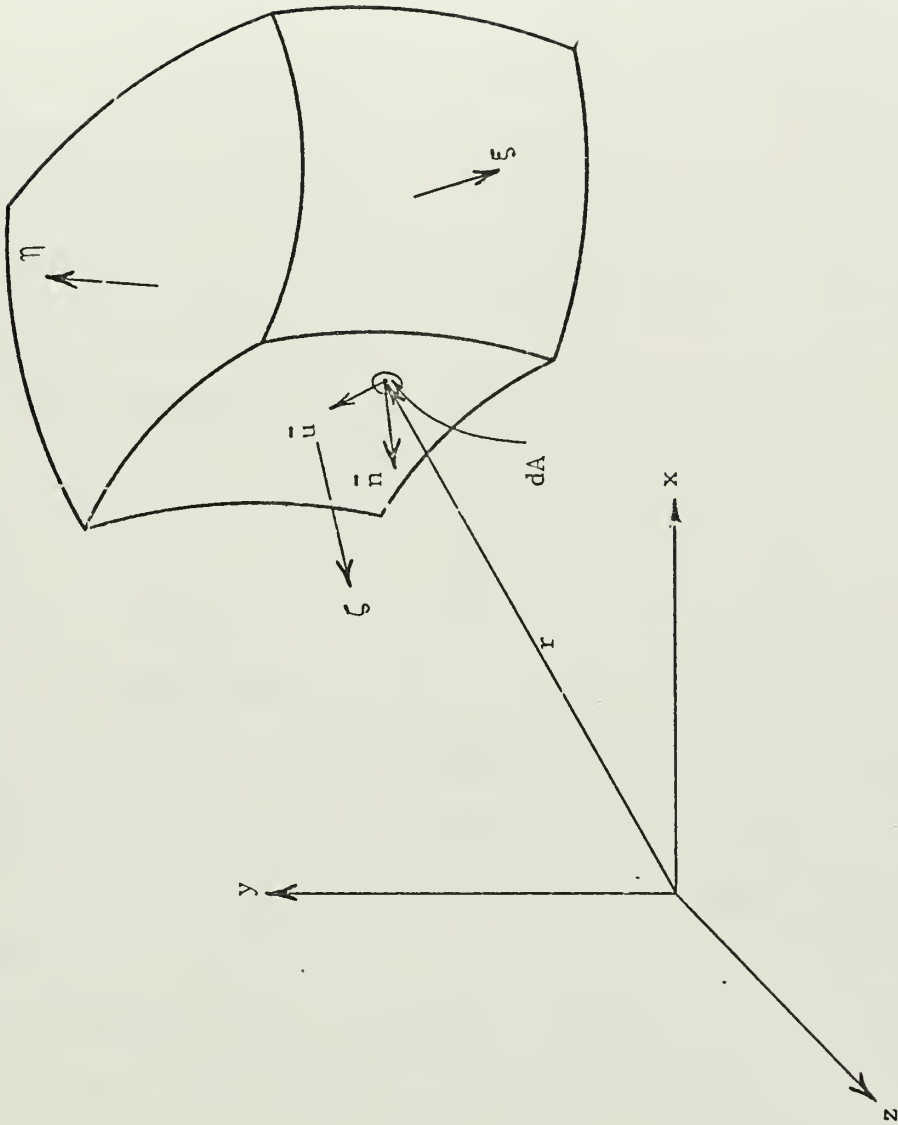


Figure 6. Pressurized Element

The integration is performed by Gaussian quadrature and the resultant vector is exhibited in convenient matrix form. Five Gauss points are used in both QUAMEG and CUMEG.

The Jacobian matrix and shape functions are determined in Subroutine GRAP by using the same function statements as are used in the calculation of the consistent gravity load vector. The face number of the element on which the pressure is acting must be included in the input data. With this information available, a "computed go to" statement is used in the code to determine which variables (ξ , η , ζ) will replace α and β . Referring to Figure 6, the faces are located as described below.

<u>Face</u>	<u>Location</u>	<u>α</u>	<u>β</u>	<u>Face</u>	<u>Location</u>	<u>α</u>	<u>β</u>
1	$\xi = +1$	η	ζ	4	$\xi = -1$	ζ	η
2	$\eta = +1$	ζ	ξ	5	$\eta = -1$	ξ	ζ
3	$\zeta = +1$	ξ	η	6	$\zeta = -1$	η	ξ

H. PLOT OF STRUCTURE MESH

Once the coordinates of all nodes in the mesh have been established, the mesh must be inspected to determine if it is acceptable. Without a plot of the coordinates, this task is nearly impossible. Plotting the mesh by hand is prohibitively time-consuming. Subroutine GRID has been included in QUAMEG and CUMEG to generate an off-line printer plot of the mesh. The parameters required for IBM 360 source library program DRAW are determined in GRID. DRAW is called to perform the actual plotting of the mesh. GRID generates a two-dimensional plot of the mesh on slices at constant values of ζ' . For the example shown in Figure 3, faces 1 and 2 and five slices at constant values of ζ' would be plotted.

The user can specify that the scaling be performed automatically by DRAW. However, that choice could result in the x-scale being not equal to the y-scale. The plot of the mesh would be distorted in that case. The scaling is done in GRID to insure that the scales will be equal and that the smallest possible scale is used.

The plotting space available with DRAW is fifteen inches in the y-direction and nine inches in the x-direction. Recall that the convention for numbering the nodes usually results in having the narrowest side of the structure along the y-axis (section 3.A). Therefore, GRID interchanges the x- and y-coordinates of the structure, for plotting purposes only, to take advantage of the larger plotting space in the y-direction. This further reduces the scale of the plot.

IV. INPUT DATA PREPARATION

This chapter is written as a self-contained unit giving specific information necessary for the preparation of input data cards. The chapter may be removed and used as a guideline for input data preparation after a thorough knowledge of the computer programs has been obtained.

A. VARIABLES

The variables and formulas defined below are included in the code and should be used as an aid for determining the arrangement of nodes within the mesh when preparing input data.

<u>Variable</u>	<u>Meaning</u>
1. NEL	Total number of elements.
$NEL = (NX)(NY)(NZ), \text{ where } NX, NY \text{ and } NZ \text{ are the number of elements along the } \xi', \eta', \text{ and } \zeta' \text{ directions} \quad (21)$	
2. NPS	Total number of nodes on any face at constant ζ' , where a face is a slice which intersects both corner nodes and mid-side nodes.
$NPS = 3(NX)(NY) + 2(NX + NY) + 1 \quad (\text{QUAMEG}) \quad (22)$	
$NPS = 5(NX)(NY) + 3(NX + NY) + 1 \quad (\text{CUMEG}) \quad (23)$	
3. NPM	Total number of mid-side nodes between any two faces.
$NPM = (NX + 1)(NY + 1) \quad (\text{QUAMEG}) \quad (24)$	
$NPM = 2(NX + 1)(NY + 1) \quad (\text{CUMEG}) \quad (25)$	
4. NPL	Total number of nodes in one layer, where a layer consists of one face plus the slice intersecting mid-side nodes between that face and an adjacent face.
$NPL = NPM + NPS \quad (26)$	
5. NUMNP	Total number of nodes in the mesh.
$NUMNP = (NZ)(NPL) + NPS \quad (27)$	

B. DATA CARDS

1. Element Field Data (7I10,2I5)

Number of cards per problem - one

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 10	NX	Number of elements in the ξ' -direction;
11 - 20	NY	Number of elements in the η' -direction;
21 - 30	NZ	Number of elements in the ζ' -direction;
31 - 40	NDX	Number of segments in the ξ' -direction;
41 - 50	NDY	Number of segments in the η' -direction;
51 - 60	NDZ	Number of segments in the ζ' -direction;
61 - 70	NPDP	Total number of nodes for which the coordinates must be specified to define the location of segment end points or other boundary nodes;
71 - 75	KORD	Specifies whether rectangular or cylindrical coordinates are used in the problem. A positive integer indicates cylindrical coordinates and a blank or zero specifies rectangular coordinates;
76 - 80	NCARD	A positive integer indicates that punched cards are desired as output. A blank or zero indicates that cards are not desired.

2. Problem Identification (10A8)

Number of cards per problem - one

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 80	TITLE	Alphanumeric information to identify the problem.

3. Nodal Point Coordinates (1I10, 3F10.0)

Number of cards per problem - one for each NPDP point

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 10	I	Nodal point number;
11 - 20	COORD(I,1)	x-coordinate (radius);

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
21 - 30	COORD(I,2)	y-coordinate (angle);
31 - 40	COORD(I,3)	z-coordinate.

4. Segment Divisions (5I10)

Number of cards per problem - one for each segment

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 10	I	Lowest node number in the segment;
11 - 20	J	Highest node number in the segment;
21 - 30	N1	Segments in the ξ' - and ζ' -directions: Number of elements above the segment (in the positive η' -direction); Segments in the η' -direction: Number of elements within the segment;
31 - 40	M1	Segments in the η' - and ζ' -directions: Number of elements to the left of the segment (in the negative ξ' -direction); Segments in the ξ' -direction: Number of elements within the segment;
41 - 50	L1	Segments in the ξ' - and η' -directions: Number of elements before the segment (in the positive ζ' -direction); Segments in the ζ' -direction: Number of elements within the segment.

5. Load Data (2I10, 2F10.0)

Number of cards per problem - one

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 10	MAP	A positive integer indicates that a plot of the structure mesh is desired. A blank or zero indicates that a plot is not desired;
11 - 20	NEFP	Number of pressurized element faces;
21 - 30	SGZ	Specific weight;
31 - 40	UDP	Uniformly distributed pressure.

6. Pressure Loading Data (2I10)

Number of cards per problem - one for each NEFP

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 10	NELP	Element number on which a pressure exists;
11 - 20	NFACE	Face on which the pressure is applied. (See section 3.G and Figure 6)

7. Plot Title Cards (6A8)

Number of cards per problem - two per plot (NZ + 1 plots)

<u>Columns</u>	<u>Variable</u>	<u>Meaning</u>
1 - 48	ITITLE	Alphameric information to identify the plot;
1 - 48	ITITLE	Alphameric information to identify the programmer and his computer center box number.

8. Stop Trap

The stop trap employed in the programs allows the execution of as many separate problems as may be desired with one submission of the program. The program will continue processing problems until it is informed that NX is less than zero. Therefore, the last card of a complete input data deck should have a negative integer in columns 1 - 10.

V. CONCLUSIONS

The addition of QUAMEG and CUMEG to the tools available in the finite element method of analysis opens new doors for the analyst who uses isoparametric finite elements. Previously, the amount of input data required to analyze a structure of appreciable size was too overwhelming for the analyst to undertake the problem. QUAMEG and CUMEG make it practical for the analyst to solve such a problem. In addition, he can be certain, after an acceptable mesh is generated, that the output data generated by QUAMEG and CUMEG is free of error.

For the example discussed in Appendix A, the number of input cards required by QUAMEG versus the number of output cards generated by QUAMEG for direct input to TRISOP is about six per cent. For the $1 \times 8 \times 8$ mesh of the pinched cylinder problem discussed in Appendix B, this ratio is about 4 per cent.

With a gravity load and pressure load acting on one cubic element, it was found that QUAMEG and CUMEG produced consistent load vectors which were accurate to at least eight decimal places.

QUAMEG and CUMEG considerably reduce the effort required of the analyst to solve a problem by the finite element method, generate acceptable values for nodal point coordinates and compute accurate consistent gravity and pressure load vectors.

VI. RECOMMENDATIONS

A. LOADING CONDITIONS

The capabilities of QUAMEG and CUMEG should be increased by adding subroutines which will generate the consistent load vectors for a thermal gradient in the structure and for an initial displacement condition at one or more nodes.

B. PLOT OF STRUCTURE MESH

It would be preferred to obtain a more complete representation of the structure mesh than is now provided. Two courses of action are recommended:

1. Expand Subroutine GRID in such a way that plots of faces in the η' - ζ' and/or ξ' - ζ' planes are also generated.

2. Replace GRID with a subroutine which would generate a three-dimensional plot of the structure mesh. The feasibility of using the IBM 360 source library program PLOT PACKAGE, which is mentioned in Ref. 12, should be investigated if this method is attempted.

This author feels that the results of recommendation 2 would be preferred to 1, although the coding may be more difficult.

APPENDIX A GENERATING A MESH WITH QUAMEG

This appendix describes the preparation of input cards and the expected results for a sample problem using QUAMEG. This problem was chosen as an example to demonstrate the capabilities of QUAMEG and to more fully explain the procedures described in Chapter 4.

The structure considered is a cylindrical solid with a star shaped void in the center. This void linearly converges to a point. Figure 7 shows a 45° sector of the structure. It is normally preferable to work with angles in the first quadrant. Therefore, the sector from zero to forty-five degrees will be considered.

The first step in the problem is to decide upon a mesh size. In this case, an $8 \times 5 \times 4$ mesh was chosen. The amount of control to be exercised by the programmer to insure that the structure will be modeled accurately can be determined now. The coordinates of all nodes along the edges of the bounding surfaces of the structure must be available to the system before the remaining nodal coordinates can be determined. Since there are no irregularities on the boundaries between the front and back faces, the coordinates of the nodes on the edges between these two faces can be computed by providing the necessary segment data. Therefore, only the nodes on the front and back faces need be considered.

At this point, two rectangles of constant ζ' are drawn with the proposed mesh superimposed upon them (see Figures 8 and 9). These figures will serve as an aid for numbering the nodes on the

actual structure and for determining the locations of segments.

After the nodes on face one are numbered, equation 22 is solved to insure that an error has not been made in the numbering. The result of equation 22 is subtracted from that of equation 27 to determine the number of the first node on the back face. In this way, the programmer need not go through the time-consuming process of numbering all nodes in the mesh. The nodes on the back face are numbered and checked as before. Note that the 8×5 mesh is in the $\xi'-\eta'$ plane. Orienting the structure so that the 5×4 face of the mesh were in the $\xi'-\eta'$ plane would reduce the half-bandwidth of the system. However, cylindrical coordinates must be used in this problem and it would not be possible to model the structure accurately if it were oriented in that manner. Finally, a picture is drawn of the front and back faces of the structure with the desired mesh superimposed and the nodes numbered (Figures 10 and 11).

The identification of segments will now be determined. The segments will be considered in the same order as they are considered in the program. Referring to Figures 8 and 9, it is seen that the following lines are parallel to the η' -axis: 1 - 11, 137 - 147, 805 - 815, 941 - 951. Now referring to Figures 10 and 11, it is seen that all of these lines are of constant angle with a uniform distribution of elements within the lines. Therefore, these lines will be treated as segments and the coordinates of the nodes listed above will be included in the input data.

By this method of specifying lines, the author does not intend to imply that the nodes along the line are numbered consecutively, starting with the lower number. For example, the nodes along line

60 - 111 in Figure 8 are: 60, 67, 77, 84, 94, 101, and 111. This method of specifying lines merely lists the numbers of nodes at the ends of a line.

In order to control the departure angles of lines 79 - 77, 96 - 94, and 113 - 111, it is necessary to force line 52 - 62 nearer the x-axis than would result if an even distribution of nodes along line 1 - 137 were allowed. Therefore, an appropriate location for node 52 is chosen, the coordinates of nodes 52 and 62 are included in the input data, and line 52 - 62 is treated as a segment. Note that this line is of neither constant angle nor constant radius. However, since the line is not on the edge of a bounding surface, the fact that it will be curved will not affect the solution adversely. Similarly, the coordinates of nodes 856 and 866 will be included in the input data and that line will be a segment.

The same rules govern the selection of segments in the ξ' -direction. Referring to Figures 8 and 9, it is seen that the coordinates of nodes along lines 1 - 137, 11 - 147, 805 - 941, and 815 - 951 must be made available to the computer. In Figures 10 and 11, line 1 - 137 is a line of constant radius; however, the element divisions along this line are not uniform. Two segments must be specified: lines 1 - 52 and 52 - 137. The coordinates of the end nodes on these segments have been specified already and need not be repeated. Line 11 - 62 is of constant radius and is therefore one segment. Again, the coordinates of these nodes have been specified previously. Line 62 - 113 is of constant angle, meeting the criterion for a segment. The coordinates of node 113 must be specified. Line 113 - 147 is of neither constant angle nor constant radius. Therefore, it cannot

be treated as a segment and the coordinates of all nodes along this line must be included in the input data. In order to further control the departure angle of the lines mentioned above, line 60 - 145 is also included as a segment. The coordinates of the end nodes of this segment have been computed in the problem already and do not need to be included in the input data. As with line 52 - 62, the exact shape of line 60 - 145 is not important. Similarly, on the back face, lines 805 - 856, 856 - 941, and 864 - 949 are treated as segments. Although the nodes along line 815 - 951 are at the origin, care must be taken to insure that the coordinates of these nodes are specified at appropriate angles in such a way that this point will have the same "shape" as the front face. The radial component of these coordinates is zero. This care is necessary because all computations in the problem are made with cylindrical coordinates. Were the coordinates of the nodes along this line specified at a radius of zero and an angle of zero, the mesh would be distorted. The following lines can also be treated as segments: 815 - 866 and 866 - 917. The coordinates of nodes 923, 934, and 940 must be included in the input data for the same reason as was given for nodes 119, 130, and 136.

To complete the process of specifying the coordinates of nodes along the structure edges, the following lines will be treated as segments in the ζ' -direction: 1 - 805, 52 - 856, 137 - 941, 60 - 864, 77 - 881, 94 - 898, 11 - 915, 11 - 815, 62 - 866, 113 - 917, and 130 - 934.

The procedure for determining the values of $N1$, $M1$, and $L1$ for the segment division data will not be discussed here. It is

felt that the explanation in Chapter 4, combined with the listing of the input cards for this problem, is sufficiently clear.

The card containing the element field data is prepared after the above steps have been accomplished. Loading information and the request for punched cards should not be included in the input data until it has been determined that the mesh will be satisfactory.

Approximately 45 seconds of computer time were required to solve this problem with QUAMEG. That time includes the 31 seconds required to compile the program. Figures 12, 13, 14, 15, and 16 are the plots of the structure mesh that was generated by QUAMEG. Table I is the listing of the required input cards described above. Figure 17 is the result of a $5 \times 5 \times 3$ mesh generated with CUMEG for the same structure. Intermediate slices and the back surface are not shown.

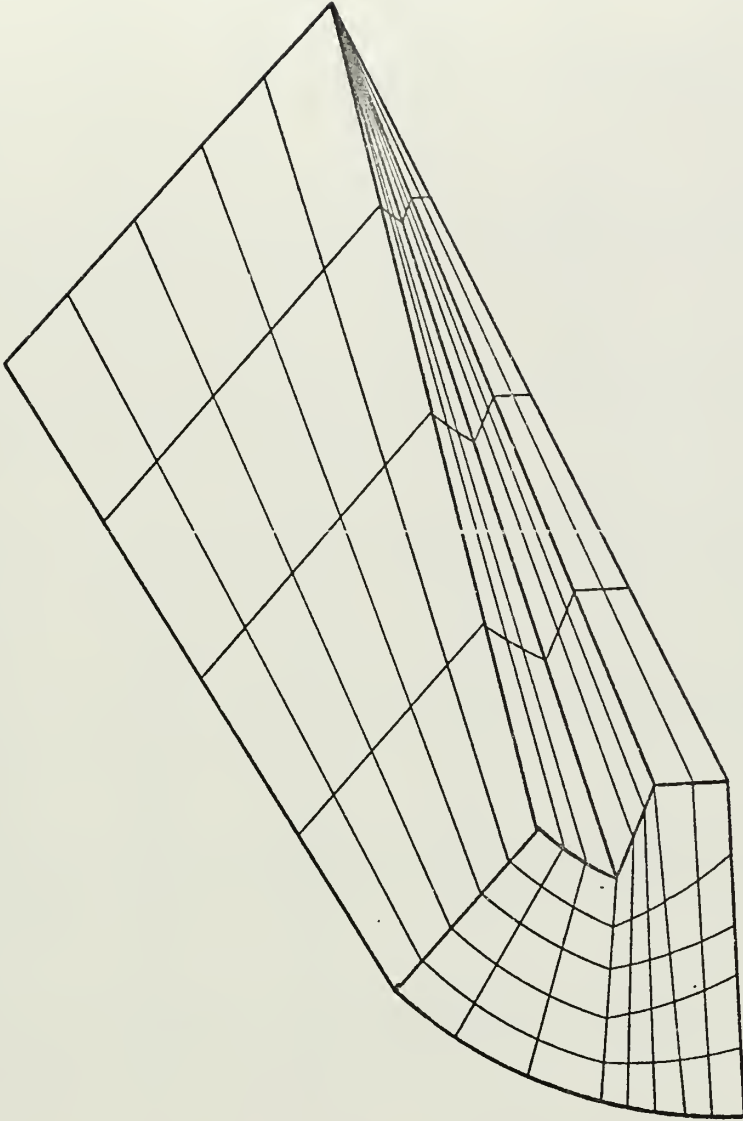


Figure 7. Problem Structure

1	52	137
11	62	147
	79	136
	96	130
	77	113
	94	111
	60	

Figure 8. Front Face

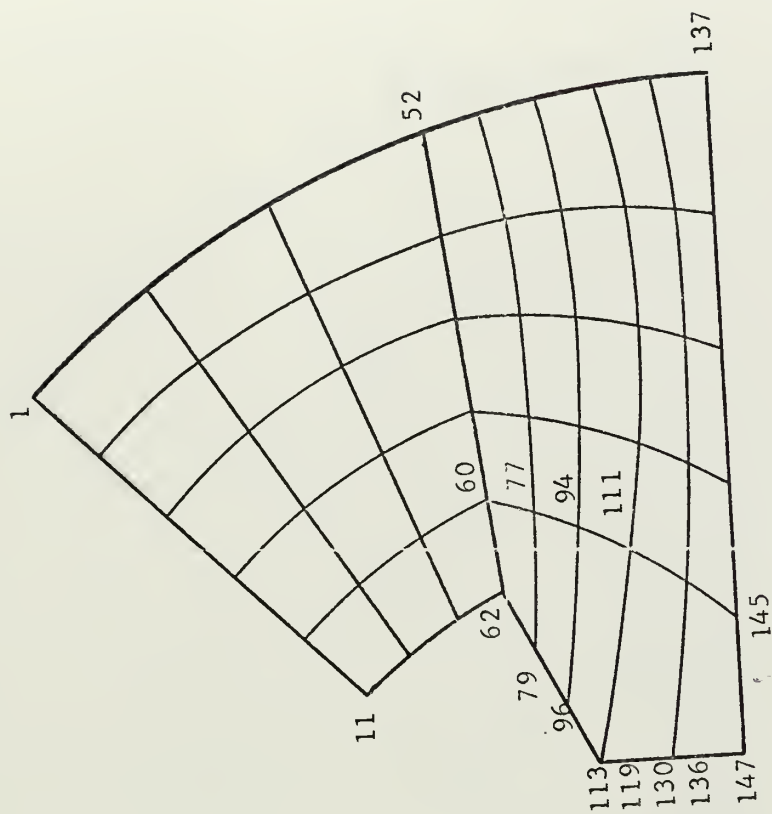


Figure 10. Front Face of Structure

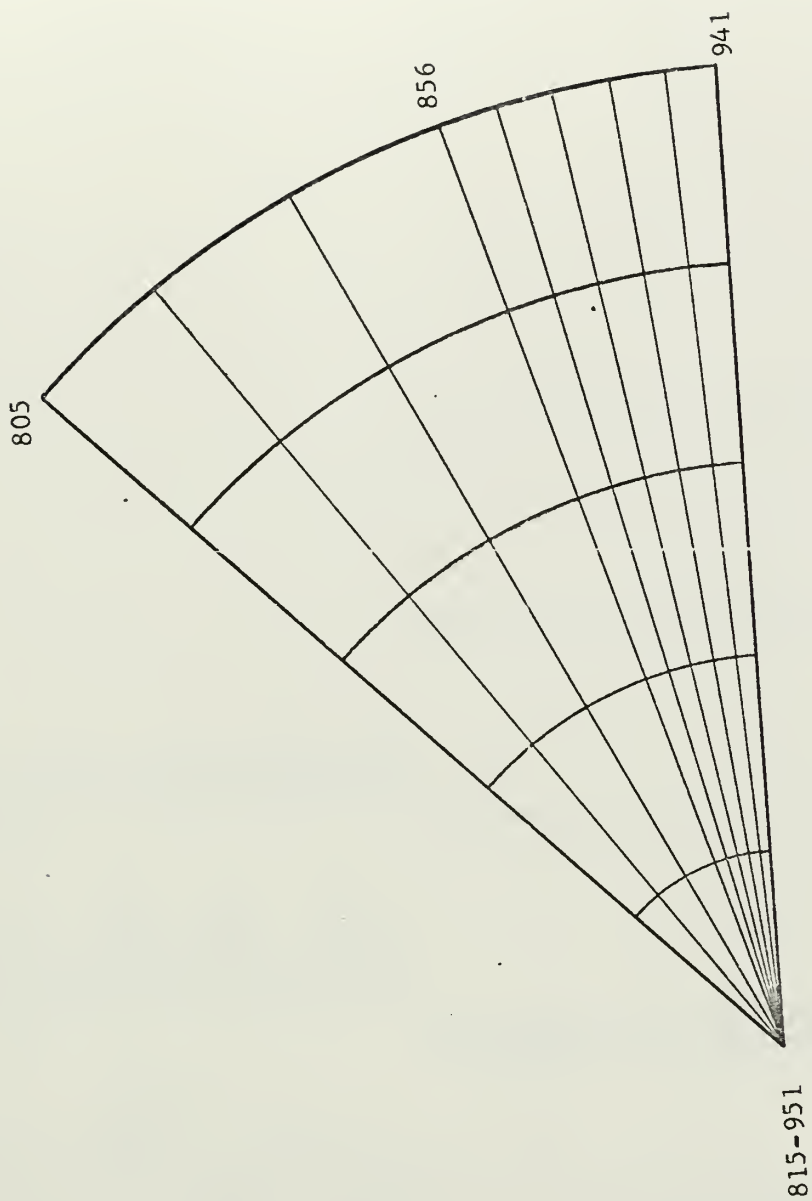


Figure 11. Back Face of Structure

Table I
Listing of Input Data Cards

1	8	5	4	10	6	12	201
	ROCKET MOTOR	PROPELLENT	MESH (8X5X4)				
15.0	45.0	5.0	5.0	0.0	0.0		
112.677	45.0	5.0	5.0	0.0	0.0		
162.677	26.565	5.0	5.0	0.0	0.0		
1131.677	26.0	5.0	5.0	0.0	0.0		
1471.0	17.0	5.0	5.0	0.0	0.0		
1525.0	10.0	5.0	5.0	0.0	0.0		
1375.0	20.556	5.0	5.0	0.0	0.0		
1191.6	24.036	5.0	5.0	0.0	0.0		
11301.5	17.125	5.0	5.0	0.0	0.0		
1361.5	45.10	5.0	5.0	0.0	0.0		
18150.0	26.565	5.0	5.0	0.0	0.0		
8660.0	26.565	5.0	5.0	0.0	0.0		
9170.0	45.0	5.0	5.0	0.0	0.0		
8055.0	17.0	5.0	5.0	0.0	0.0		
8565.0	10.0	5.0	5.0	0.0	0.0		
9415.0	20.556	5.0	5.0	0.0	0.0		
9230.0	24.036	5.0	5.0	0.0	0.0		
9340.0	17.125	5.0	5.0	0.0	0.0		
9510.0	7.0	5.0	5.0	0.0	0.0		
1	11	5	5	5	0	0	0
527	627	147	55	137	5	5	5
1375	1475	189	55	137	5	5	5
856	1475	89	55	137	5	5	5
856	1475	89	55	137	5	5	5
1	1475	89	55	137	5	5	5
520	1475	89	55	137	5	5	5
60	1475	89	55	137	5	5	5
1125	1475	89	55	137	5	5	5
8055	1475	89	55	137	5	5	5
815	1475	89	55	137	5	5	5
8664	1475	89	55	137	5	5	5
8664	1475	89	55	137	5	5	5
1	1475	89	55	137	5	5	5
527	1475	89	55	137	5	5	5
1375	1475	89	55	137	5	5	5
1130	1475	89	55	137	5	5	5
1147	1475	89	55	137	5	5	5
1	1475	89	55	137	5	5	5
60	1475	89	55	137	5	5	5

4
4

5
6

4
4 0.0

94
111
111
898
915
MESH II FACE 1
HANSON II BOX 15 2
MESH II BOX 15 3
MESH II BOX FACE 4
HANSON II BOX FACE 5
MESH II BOX FACE
HANSON II BOX 15
MESH II BOX FACE
HANSON II BOX 15

-111111

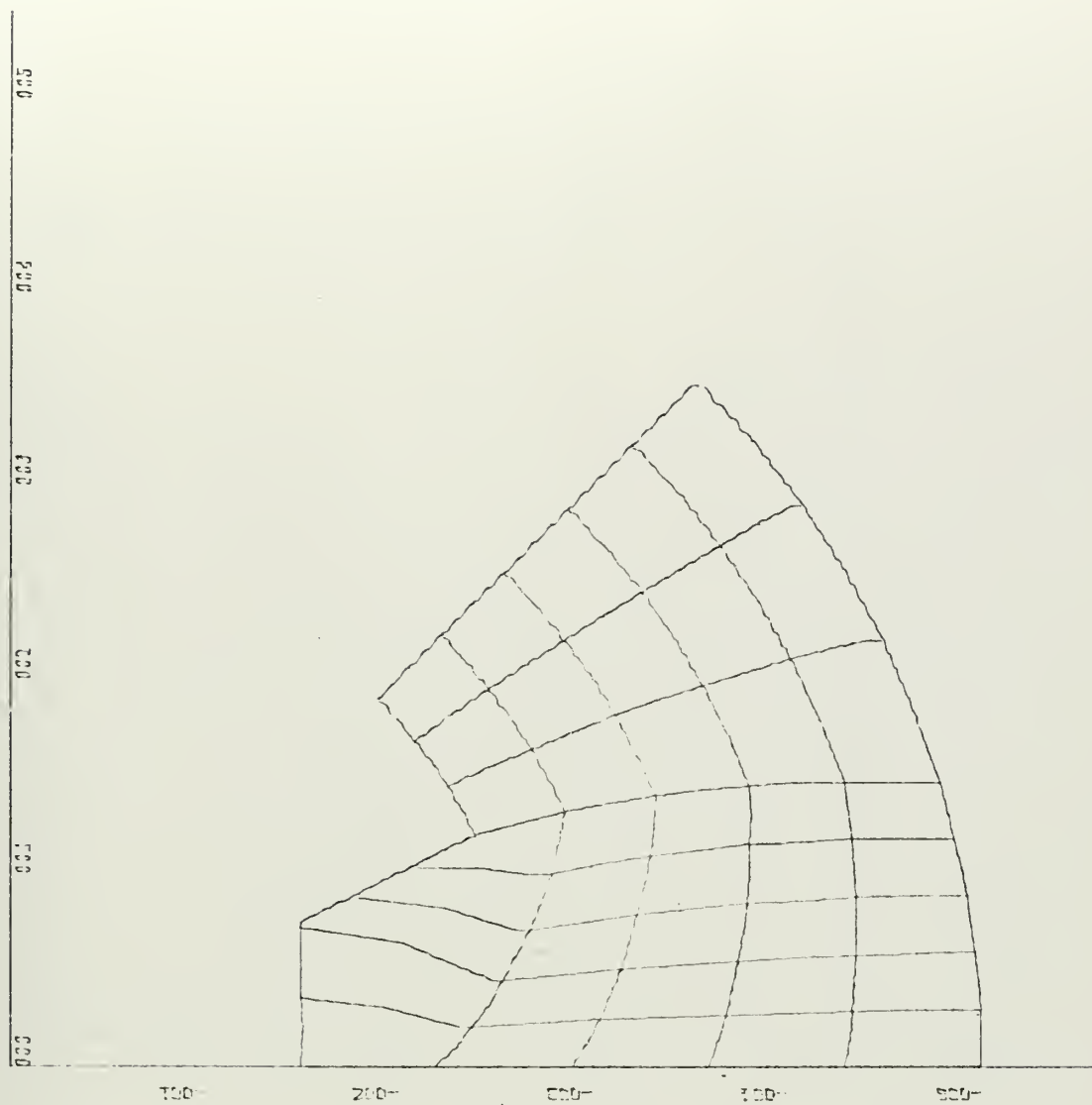


Figure 12. Generated Mesh For Front Face

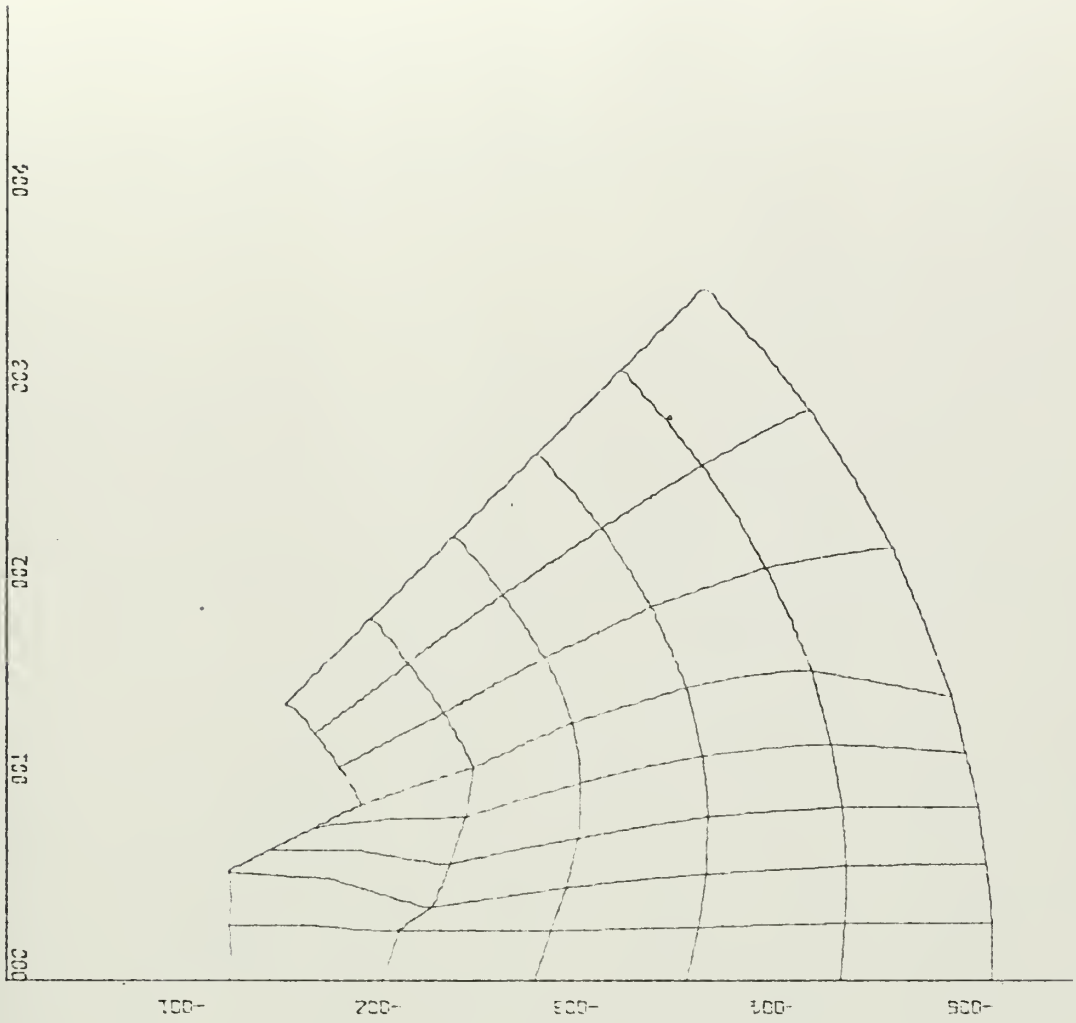


Figure 13. Generated Mesh For First Intermediate Slice

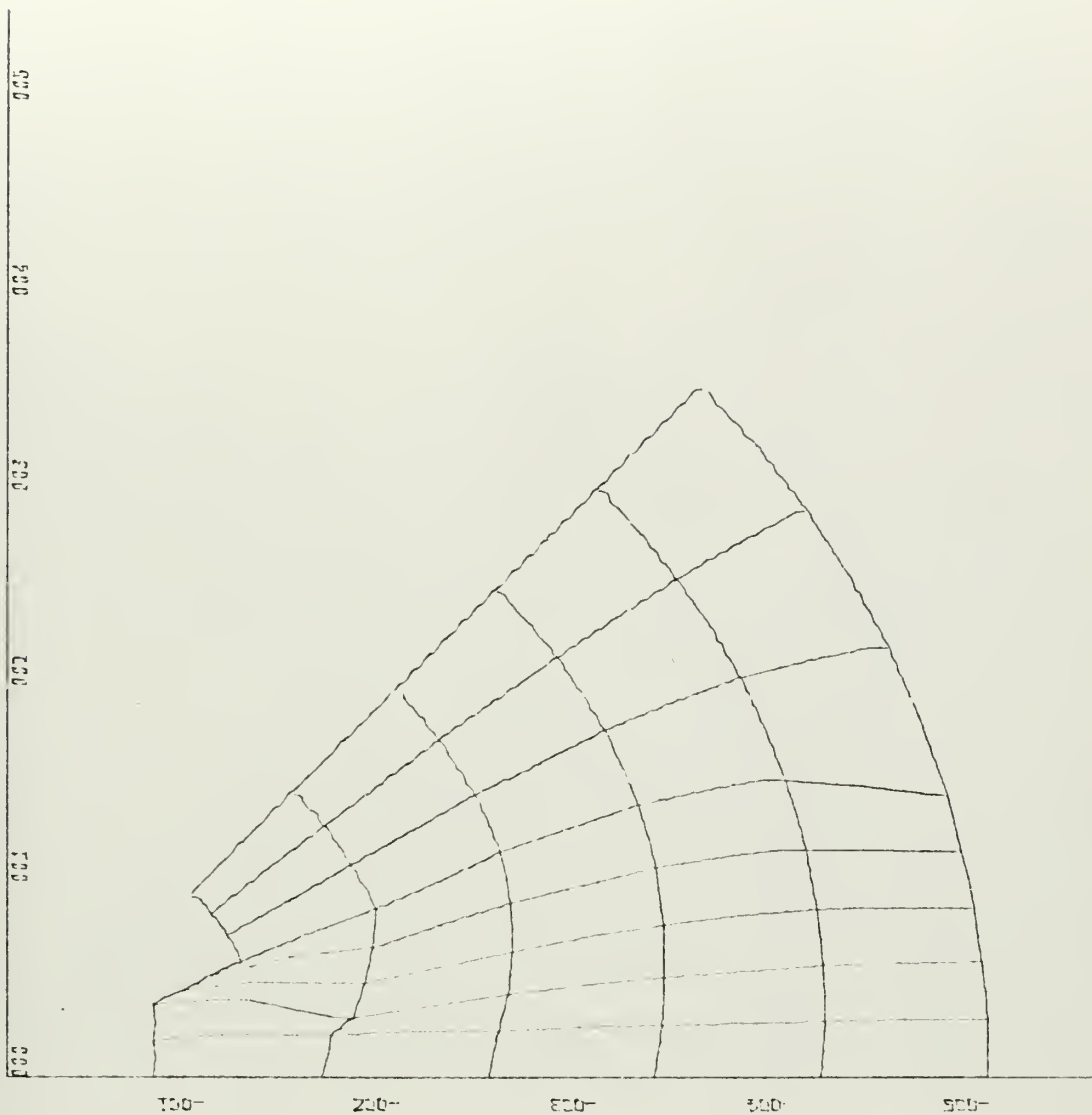


Figure 14. Generated Mesh For Second Intermediate Slice

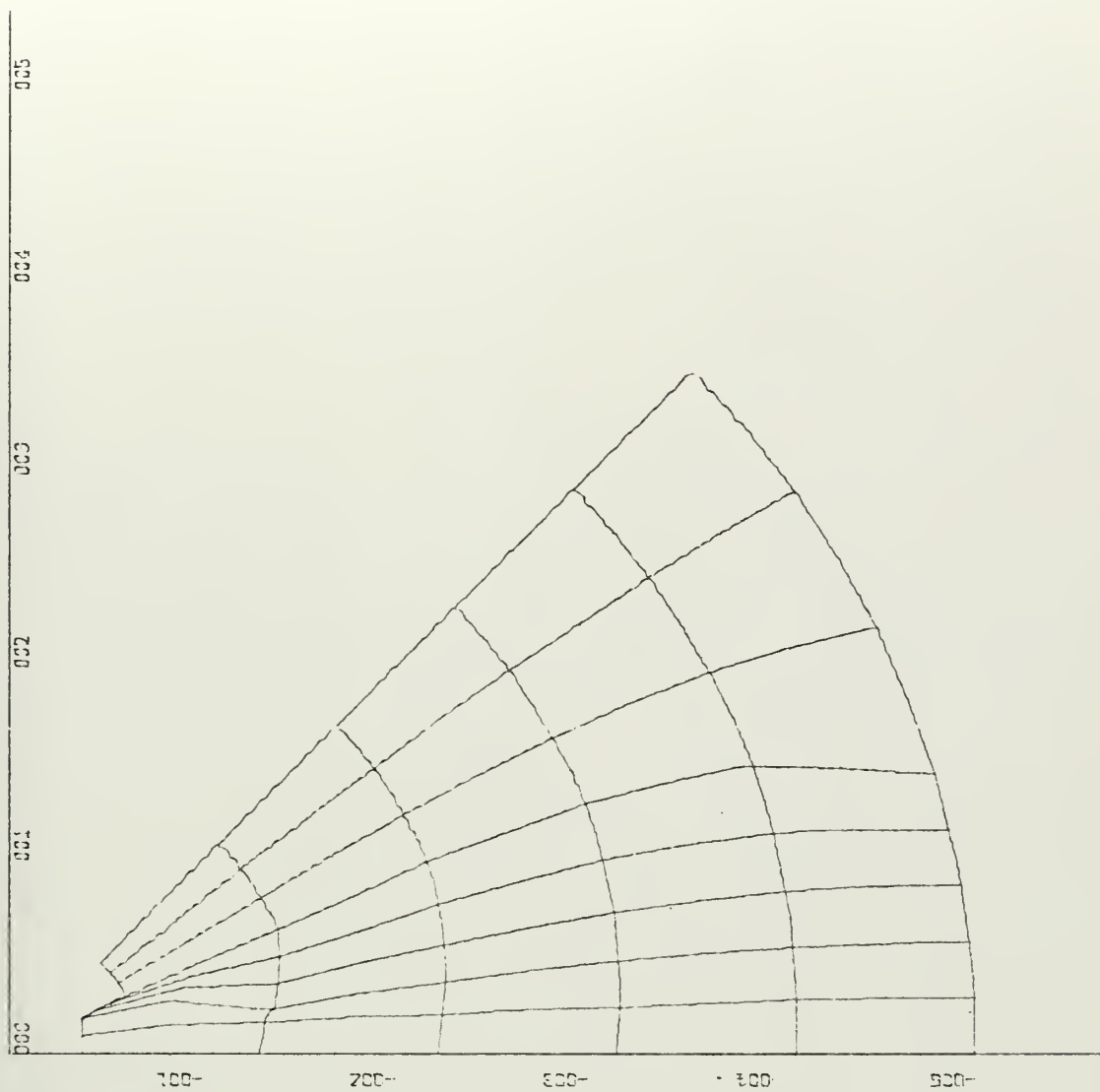


Figure 15. Generated Mesh For Third Intermediate Slice

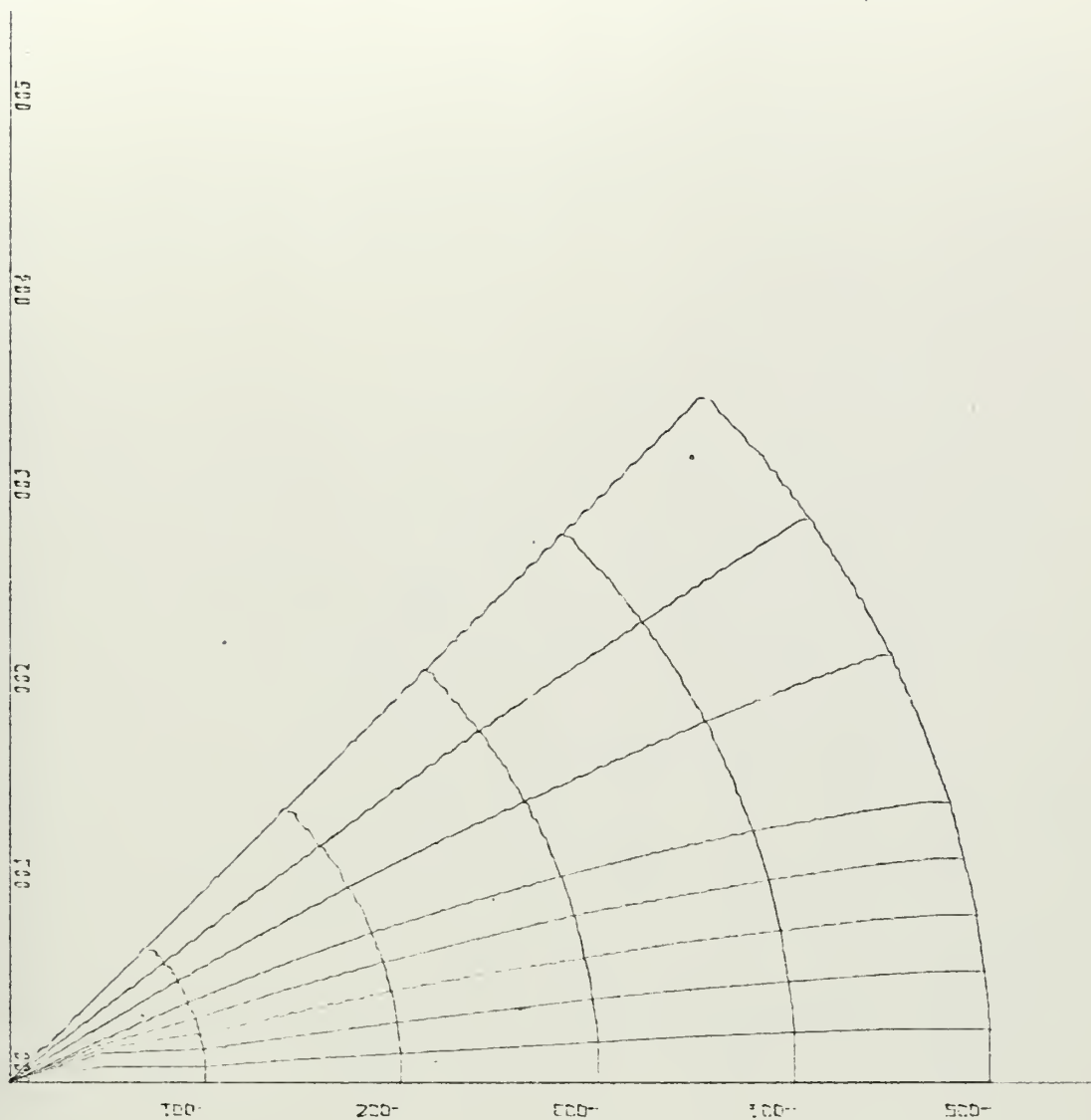


Figure 16. Generated Mesh For Back Face

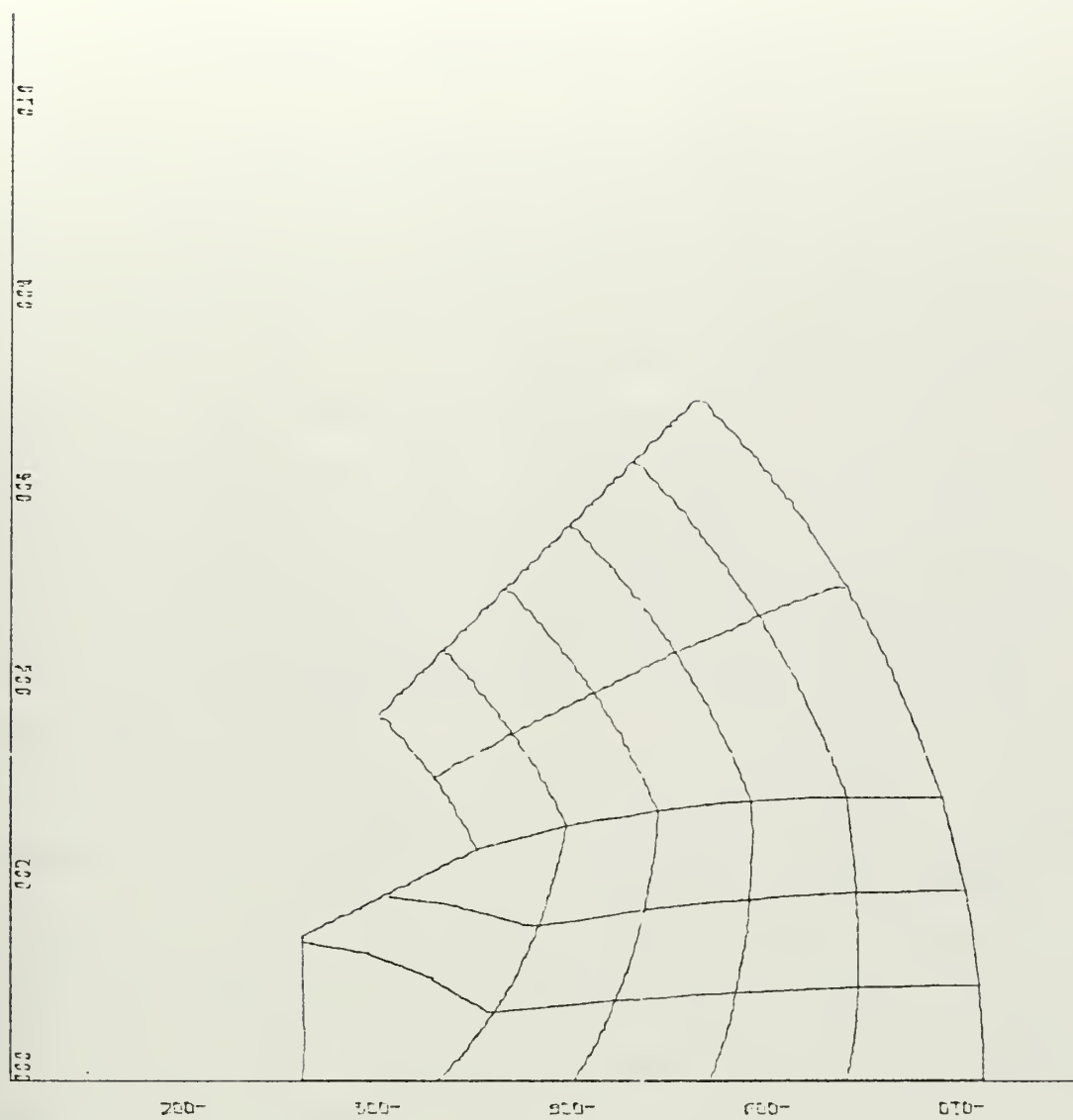


Figure 17. Generated Mesh For Front Face

APPENDIX B

A CLASSICAL PROBLEM SOLVED WITH TRISOP

TRISOP is a relatively new computer system which uses isoparametric finite elements to obtain stresses, strains, and displacements of three-dimensional structures that are subjected to static loads.

Although a number of test problems have been conducted with TRISOP, the input preparation was too voluminous to contemplate a full evaluation of the system. Without QUAMEG, the difficulties discussed in Chapter 1 limited the size and number of problems that could be solved. In order to evaluate TRISOP as a complete system, a classical problem was selected and a convergence study was made, using QUAMEG and TRISOP.

The problem selected was that of the pinched, thin-shell cylinder. Obviously, this problem should be solved using thin-shell elements for maximum efficiency. The purpose of this test was to determine if the three-dimensional solution would converge to the same results as are reported in Ref. 3 and also to determine how well thin-shell action is captured by TRISOP. The results reported by Cantin [3] are included in Table II with the results obtained in the test. The problem is described in Figure 18.

Conclusions:

Although the results of this convergence study indicate a tendency to approach the same results obtained by thin-shell theory, acceptable results can be obtained only with a very fine mesh.

TRISOP can be used to solve problems for which there are no closed-form or accurate numerical methods of solution available. However, the user must be prepared to pay the price of long computational times, for some problems.

Table II
Displacement at the applied load for a pinched cylinder.

Bogner et al.			Cantin		
No. of Eq.	Mesh	Displace., in.	No. of Eq.	Mesh	Displac., in.
72	1 x 2	-0.0802	36	1 x 2	-0.0921
96	1 x 3	-0.1026	48	1 x 3	-0.1072
120	1 x 4	-0.1087	60	1 x 4	-0.1099
108	2 x 2	-0.0808	54	2 x 2	-0.0931
144	2 x 3	-0.1036	72	2 x 3	-0.1085
180	2 x 4	-0.1098	90	2 x 4	-0.1113
			150	4 x 4	-0.1126
			294	6 x 6	-0.1137
			486	8 x 8	-0.1139
			726	10 x 10	-0.1139

Hanson

No. of Eq.	Mesh	Displac., in.
153	1 x 2 x 2	-0.00177
243	2 x 2 x 2	-0.00178
423	2 x 2 x 4	-0.00196
465	1 x 4 x 4	-0.00375
945	1 x 6 x 6	-0.03002
1593	1 x 8 x 8	-0.08702

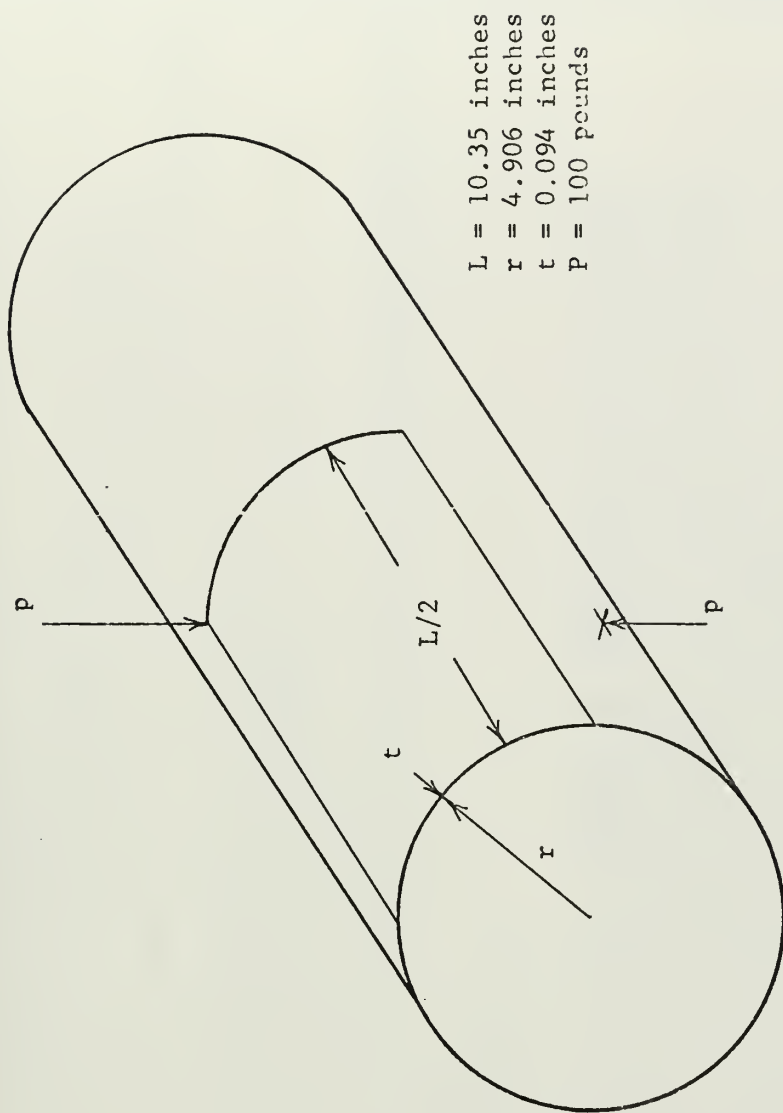


Figure 18. Pinched Cylinder

APPENDIX C COMPUTER LISTING (QUAMEG)

```

*****
**      MESH GENERATING PROGRAM FOR TRISOP
**      USING QUADRATIC ELEMENTS
**      CODED BY D. E. HANSON, FEBRUARY 1971
**      FRONT FACE IS IN XY PLANE, LEFT FACE IS IN YZ PLANE
**      NODAL POINT ONE IS AT UPPER LEFT CORNER OF FRONT FACE
*****

```

```

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION GLV(999),PLM(999,3),NELC(8)
DIMENSION NCON(200,21),KODE(999),NC(6,8),TITLE(10)
COMMON /NB1/ NPEL,NFACE,NUMNP
COMMON /B2/ COORD(999,3),COREL(20,3)
COMMON /B3/ AK(20),AK1(20)
COMMON /B4/ AK2(8,3),AK3(8,3)
DATA NC/13,13,1,3,17,15,9,14,2,10,18,14,1,15,3,15,19,13,8,10,4,16,
112,20,7,3,5,17,7,19,12,2,6,11,6,18,19,1,7,5,5,17,20,9,8,4,11,16/
1000 FORMAT(11,)
1100 FORMAT(7I10,2I5)
1200 FORMAT(5I10)
1300 FORMAT(11I10,3F10.0)
1400 FORMAT(1X,2G13,1I5)
1500 FORMAT(///,
115X,NELC(11),
215X,NUMNP,
315X,NFACE,
415X,NBAND,
515X,NBAND,
615X,NBAND,
715X,NBAND,
815X,NBAND,
915X,NBAND,
115X,NBAND,
1215X,NBAND,
136X,CONNECTIVITY MATRIX'//,
211X,ELC(11),
3S NFACE,
FORMAT(11X,14,2X,20I3,15)
1700 FORMAT(6X,110,3F15.5)
1800 FORMAT(21I10,2F10.0)
1900 FORMAT(///35X,COORDINATES'//,14X,NP,9X,'X',14X,'Y',14X,'Z')
2000 FORMAT(///35X,GRAVITY LOAD'//,14X,NP,11X,'Z')
2100 FORMAT(///20X,PRESSURE INTENSITY ='F5.2, //27X,EL',5X,'FACE')
2200

```



```

2300 FORMAT(6X,I10,F20.12)
2400 FORMAT(I10,20X,F20.12)
2500 FORMAT(///,30X,'PRESSURE LOADS',//,14X,'NP', 9X,'X',14X,'Y',14X,
1 1Z',//)
2600 FORMAT(I10,3F20.12)
2700 FORMAT(2I10)
2800 FORMAT(10A8)
2900 FORMAT(24X,I5,7X,I1)
10 CONTINUE
READ(5,1100) NX,NY,NZ,NDX,NDY,NDZ,NPDP,KORD,NCARD
IF(NX.LT.0) STOP
READ(5,2800) TITLE
WRITE(6,2800) TITLE
PI=3.1415926536DC/180.0DC
ZRO=0.0DC
ORF=1.8DC
FP=4.0DC
NEL = NX*NY*NZ
NX1=NX-1
NZ1=NZ-1
NY1 = NY+1
NY2 = 2*NY
NY3 = 3*NY
NY4=NY2+1
NY5=NY3+2
NY6=NY-1
NY7=NY2-1
NPS=NX*NY5+NY4
NPM=NY1*(NX+1)
NPL = NPS + NPM
NUMNP = NZ*NPL + NPS

```

C
C
C

COMPUTE ELEMENT CONNECTIVITY

```

DO 15 IN=1,NZ
IZ = IN-1
DO 15 IR=1,NX
IM = IR-1
NR = NX-IM
IC = IM*NY
I1 = I1+1
I2 = I2+1
I3 = I3+1
I4 = I4+1
I5 = I5+1
I6 = I6+1
I7 = I7+1
I8 = I8+1

```


QMG01730
 QMG01740
 QMG01750
 QMG01760
 QMG01770
 QMG01780
 QMG01790
 QMG01800
 QMG01810
 QMG01820
 QMG01830
 QMG01840
 QMG01850
 QMG01860
 QMG01870
 QMG01880
 QMG01890
 QMG01900
 QMG01910
 QMG01920
 QMG01930
 QMG01940
 QMG01950
 QMG01960
 QMG01970
 QMG01980
 QMG01990
 QMG02000
 QMG02010
 QMG02020
 QMG02030
 QMG02040
 QMG02050
 QMG02060
 QMG02070
 QMG02080
 QMG02090
 QMG02100
 QMG02110
 QMG02120
 QMG02130
 QMG02140
 QMG02150
 QMG02160
 QMG02170
 QMG02180
 QMG02190
 QMG02200

```

COORD(L,1)=COORD(I,1)+K*DX
COORD(L,2)=COORD(I,2)+K*DY
COORD(L,3)=COORD(I,3)+K*DZ
CONTINUE
70 IF(NDX.EQ.0) GO TO 120
80 DO 110 LA=1,NDX
  READ(5,1200) I,J,N1,M1,L1
  MD=2*M1
  DX=(COORD(J,1)-COORD(I,1))/MD
  DY=(COORD(J,2)-COORD(I,2))/MD
  DZ=(COORD(J,3)-COORD(I,3))/MD
  DO 110 K=1,M1
    K1=K-1
    K2=2*K
    K3=K2-1
    L=I+K*NY5
    L2=I+NY4+K1*NY5-N1
    KL=CODE(L)
    KL2=CODE(L2)
    IF(KL2.GT.0) GO TO 90
    CODE(L2)=1
    COORD(L2,1)=COORD(I,1)+K3*DX
    COORD(L2,2)=COORD(I,2)+K3*DY
    COORD(L2,3)=COORD(I,3)+K3*DZ
    IF(KL.GT.0) GO TO 110
    CODE(L)=1
    COORD(L,1)=COORD(I,1)+K2*DX
    COORD(L,2)=COORD(I,2)+K2*DY
    COORD(L,3)=COORD(I,3)+K2*DZ
    CONTINUE
  110 IF(NDX.EQ.0) GO TO 150
  120 DO 140 LA=1,NDZ
    READ(5,1200) I,J,N1,M1,L1
    LD=2*L1
    DX=(COORD(J,1)-COORD(I,1))/LD
    DY=(COORD(J,2)-COORD(I,2))/LD
    DZ=(COORD(J,3)-COORD(I,3))/LD
    DO 140 K=1,L1
      K1=K-1
      K2=2*K
      K3=K2-1
      L=I+K*NPL
      L2=I+NX*NPL
      KL=CODE(L)
      KL2=CODE(L2)
      IF(KL2.GT.0) GO TO 130
      CODE(L2)=1
      COORD(L2,1)=COORD(I,1)+K3*DX

```


QMG02210
QMG02220
QMG02230
QMG02240
QMG02250
QMG02260
QMG02270
QMG02280

QMG02285
QMG02290
QMG02300
QMG02310
QMG02320
QMG02330
QMG02340
QMG02350
QMG02360
QMG02370
QMG02380
QMG02390
QMG02400
QMG02410
QMG02420
QMG02430
QMG02440
QMG02450
QMG02460
QMG02470
QMG02480
QMG02490
QMG02500
QMG02510

QMG02515
QMG02520
QMG02530
QMG02540
QMG02550
QMG02560
QMG02570
QMG02580

```

130 COORD(L2,2)=COORD(I,2)+K3*DX
      COORD(L2,3)=COORD(I,3)+K3*DZ
      IF(KL.GT.0) GO TO 140
      KODE(L)=1
      COORD(L,1)=COORD(I,1)+K2*DX
      COORD(L,2)=COORD(I,2)+K2*DY
      COORD(L,3)=COORD(I,3)+K2*DZ
140 CONTINUE

```

C COMPUTE COORDINATES OF CORNER NODES

C FRONT AND BACK FACES

```

150 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 165
      DO 160 L1=1,2
      L2=L1-1
      DO 160 ITER=1,60
      DO 160 I=1,NX1
      IC=(I-1)*NY
      DO 160 J=1,NY6
      IL1=IC+J+(NEL-NX*NY)*L2
      IL2=IL1+1
      IL3=IL1+NY
      I1=1+12*L2
      I2=I1+4
      I3=I1+6
      JT1=NCON(I1,I3)
      IF(KODE(JT1).GT.0) GO TO 160
      JTA=NCON(IL1,I1)
      JTB=NCON(IL1,I2)
      JTC=NCON(IL2,I3)
      JTD=NCON(IL3,I3)
      DO 160 N=1,3
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
      1-COORD(JT1,N)
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX
160 CONTINUE

```

C LEFT AND RIGHT FACES

```

165 IF((NY.EQ.1).OR.(NZ.EQ.1)) GO TO 175
      DO 170 L1=1,2
      L2=2*(L1-1)
      L3=L1-1
      DO 170 ITER=1,60
      DO 170 I=1,NZ1
      IM=I-1
      DO 170 J=1,NY6

```


QMG02590
QMG02600
QMG02610
QMG02620
QMG02630
QMG02640
QMG02650
QMG02660
QMG02670
QMG02680
QMG02690
QMG02700
QMG02710
QMG02720
QMG02730
QMG02740
QMG02750

QMG02755
QMG02760
QMG02770
QMG02780
QMG02790
QMG02800
QMG02810
QMG02820
QMG02830
QMG02840
QMG02850
QMG02860
QMG02870
QMG02880
QMG02890
QMG02900
QMG02910
QMG02920
QMG02930
QMG02940
QMG02950
QMG02960
QMG02970
QMG02980

QMG02985

```

111=J+NY*(NX*(IM+L3)-L3)
112=111+1
113=111+NX*NY
11=15-L2
12=5+L2
13=17+L2
JTI=NCON(11,13)
IF(KODE(JTI).GT.0) GO TO 170
JTA=NCON(11,11)
JTB=NCON(11,12)
JTC=NCON(12,13)
JTD=NCON(13,13)
DO 170 N=1,3
DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
1-COORD(JTI,N)
COORD(JTI,N)=COORD(JTI,N)+ORF*DX
170 CONTINUE

```

C TOP AND BOTTOM FACES

```

175 IF((NX.EQ.1).OR.(NZ.EQ.1)) GO TO 185
DO 180 L1=1,2
L2=L1-1 ITER=1,60
DO 180 I=1,NZ1
DO 180 J=1,NX1
IC=(I-1)*NX*NY+L2*NY6+1
DO 180 J=1,NX1
111=IC+NY*(J-1)
112=111+NX*NY
113=111+NY
11=1+6*L2
12=15+2*L2
13=11+12
JTI=NCON(11,13)
IF(KODE(JTI).GT.0) GO TO 180
JTA=NCON(11,12)
JTB=NCON(11,11)
JTC=NCON(12,13)
JTD=NCON(13,13)
DO 180 N=1,3
DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
1-COORD(JTI,N)
COORD(JTI,N)=COORD(JTI,N)+ORF*DX
180 CONTINUE

```

C INTERIOR FACES

```

185 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 195

```



```

K=6
K1=1
DO 220 J=1,3,2
I1=J+(I-1)*12
I2=I1+K
I3=I2+K1
K=2
K1=-1
J1=NCON(L,I3)
IF(KODE(J1),GT.0) GO TO 220
J2=NCON(L,I1)
J3=NCON(L,I2)
DO 220 N=1,3
COORD(JT1,N)=COORD(JT2,N)-(COORD(JT2,N)-COORD(JT3,N))/2.0D0
CONTINUE
220 WRITE(6,2000)
DO 240 L=1,NUMNP
KODE(L)=0
IF(KORD.EQ.0) GO TO 230
PHI=PI*COORD(L,2)
RAD=COORD(L,1)
COORD(L,1)=RAD*DCOS(PHI)
COORD(L,2)=RAD*DSIN(PHI)
DO 240
IF(NCARD.EQ.0) GO TO 240
WRITE(7,1800) L,(COORD(L,LP),LP=1,3)
230 WRITE(6,1800) L,(COORD(L,LP),LP=1,3)
240 READ(5,1900) MAP,NEFF,SGZ,UDP
IF((SGZ.EQ.0.0).AND.(NEFF.EQ.0)) GO TO 280
IF(SGZ.EQ.0.0) GO TO 280

DETERMINE CONSISTENT GRAVITY LOAD

DO 260 I=1,NEL
DO 250 L=1,NPEL
J2=NCON(I,L)
DO 250 K=1,NDF
COREL(L,K)=COORD(J2,K)
CALL CUB4V
DO 260 J=1,NPEL
J1=NCON(I,J)
KODE(J1)=1
GLV(J1)=GLV(J1)+SGZ*AK1(J)
CONTINUE
260 WRITE(6,2100)
DO 270 I=1,NUMNP
IF((NCARD.EQ.0).OR.(NEFF.NE.0)) GO TO 270
WRITE(7,2400) I,GLV(I)
270 WRITE(6,2300) I,GLV(I)

```

DETERMINE CONSISTENT GRAVITY LOAD


```

C
C
C
      DETERMINE CONSISTENT PRESSURE LOAD
280  IF(NEFP.EQ.0) GO TO 330
      WRITE(6,2200) UDP
      DO 300 I=1,NEFP
      READ(5,2700) NELP,NFACE
      WRITE(6,2900) NELP,NFACE
      DO 290 J=1,8
      J1=NC(NFACE,J)
      NELC(J)=NCON(NELP,J1)
      DO 295 L=1,NPEL
      J2 = NCON(NELP,L)
      DO 295 K=1,NDF
      COREL(L,K) = COORD(J2,K)
      CALL QUAD5
      DO 300 J=1,8
      J1=NELC(J)
      KODE(J1) = 1
      DO 300 L=1,NDF
      PLM(J1,L)=PLM(J1,L)+UDP*AK3(J,L)
300  WRITE(6,2500)
      DO 320 I=1,NUMNP
      IF(KODE(I).EQ.0) GO TO 320
      IF(NCARD.EQ.0) GO TO 310
      PLM(I,3)=PLM(I,3)+GLV(I)
      WRITE(7,2600) I,(PLM(I,J),J=1,3)
      WRITE(6,1800) I,(PLM(I,L),L=1,3)
310  WRITE(6,1000)
320  CONTINUE
330  CONTINUE
      WRITE(6,1000)
C
C
C
      DRAW STRUCTURE MESH
      IF(MAP.EQ.0) GO TO 340
      CALL GRID(NX,NY,NZ,NUMNP,NPL)
      CONTINUE
      GO TO 10
340  END

```

```

QMG03900
QMG03910
QMG03920
QMG03930
QMG03940
QMG03950
QMG03960
QMG03970
QMG03971
QMG03972
QMG03973
QMG03974
QMG03980
QMG03990
QMG04000
QMG04010
QMG04020
QMG04030
QMG04040
QMG04050
QMG04060
QMG04070
QMG04080
QMG04090
QMG04100
QMG04110
QMG04120
QMG04130

QMG04140
QMG04150
QMG04160
QMG04170
QMG04180

```



```

SUBROUTINE CUB4V
IMPLICIT REAL*8(A-H,O-Z)
COMMON /NB1/ NPEL,NEL,NFACE,NUMNP
COMMON /B3/ AK(20),AKI(20)
DIMENSION AI(4),XI(4),AIA(4,4,4)
DATA XI/0.8611363115940526,0.3399810435848563,
1 DATA AI/0.3399810435848563,-0.8611363115940526/
1 DATA AI/0.3478548451374539,0.6521451548625461,
1 DO 100 I=1,NPEL
100 AK1(I)=0.0D0
DO 200 I=1,4
DO 200 J=1,4
DO 200 K=1,4
200 AIA(I,J,K)=AI(I)*AI(J)*AI(K)
DO 400 I=1,4
X=XI(I)
DO 400 J=1,4
Y=YI(J)
DO 400 K=1,4
Z=XI(K)
CALL GRAP(X,Y,Z,1)
DO 300 L=1,NPEL
300 AK1(L)=AK1(L)+AIA(I,J,K)*AK(L)
400 CONTINUE
RETURN
END

```

```

CUB00010
CUB00020
CUB00030
CUB00040
CUB00050
CUB00060
CUB00070
CUB00080
CUB00090
CUB00100
CUB00110
CUB00120
CUB00130
CUB00140
CUB00150
CUB00160
CUB00170
CUB00180
CUB00190
CUB00200
CUB00210
CUB00220
CUB00230
CUB00240
CUB00250
CUB00260
CUB00270

```



```

SUBROUTINE QUAD5
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION AIA(5,5),AI(5),XI(5)
COMMON /NB1/ NPEL,NEL,NFACE,NUMNP
COMMON /B4/ AK2(8,3),AK3(8,3)
DATA XI/O.9061798459386640,0.5384693101056831,0.000,-0.5384693101010
156831,-0.9061798459386640/
DATA AI/O.2369268850561891,0.4786286704993665,0.5688888888888889,0.5688888888888889,0.5688888888888889,0.5688888888888889/
DO 100 I=1,8
DO 100 J=1,3
100 AK3(I,J)=0.000
DO 200 I=1,5
DO 200 J=1,5
200 AIA(I,J)=AI(I)*AI(J)
A=1.000
IF(NFACE.GT.3) A=-1.000
DO 400 I=1,5
B=XI(I)
DO 400 J=1,5
C=XI(J)
GO TO (210,220,230,210,220,230),NFACE
210 X=A
Y=B
Z=C
GO TO 240
220 X=C
Y=A
Z=B
GO TO 240
230 X=B
Y=C
Z=A
240 CONTINUE
CALL GRAP(X,Y,Z,2)
DO 300 K=1,8
DO 300 L=1,3
300 AK3(K,L)=AK3(K,L)+AIA(I,J)*AK2(K,L)
400 CONTINUE
RETURN
END

```



```

SUBROUTINE GRAP(X,Y,Z,N)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /NB1/ NPEL,NFACE,NUMNP
COMMON /B2/ COORD(999,3),COREL(20,3)
COMMON /B3/ AK(20),AK1(20)
COMMON /B4/ AK2(8,3),AK3(8,3)
DIMENSION W1(3,20),W2(20),W3(8),AJ(3,3),CORDG(20,3),
1 CRP(3),NC(6,8)
DATA CORDG/1.000,0.000,-1.000,-1.000,-1.000,0.000,1.000,1.000,
1 1.000,-1.000,-1.000,1.000,0.000,-1.000,-1.000,0.000,
2 1.000,1.000,1.000,1.000,0.000,-1.000,-1.000,0.000,
3 1.000,1.000,-1.000,-1.000,1.000,0.000,-1.000,-1.000,
4 -1.000,0.000,1.000,1.000,1.000,1.000,1.000,1.000,
5 0.000,0.000,0.000,-1.000,-1.000,-1.000,-1.000,-1.000,
6 -1.000,-1.000/
DATA NC/13,13,1,3,17,15,9,14,2,10,18,14,1,15,3,15,19,13,8,10,4,16,
1 12,20,7,3,5,17,7,19,12,2,6,11,6,18,19,1,7,5,5,17,20,9,8,4,11,16/
DE(C,D,E,X1,Y1,Z1)=X1*(1.000+C*D*Y1)*(1.000+E*Z1)*(2.000*C*X1+D*Y1+
1 E*Z1-1.000)/8.000
D2(C,D,E,Y1,Z1)=-C*(1.000+D*Y1)*(1.000+E*Z1)/2.000
D4(C,A,B,C,X1,Y1,Z1)=(1.000-C*C)*Y1*(1.000+E*Z1)/4.000
SFC(A,B,C,X1,Y1,Z1)=(1.000+A*X1)*(1.000+B*Y1)*(1.000+C*Z1)*(A*X1+
1 B*Y1+C*Z1-2.000)/8.000
SEFM(A,B,C,Y1,Z1)=(1.000-A*A)*(1.000+B*Y1)*(1.000+C*Z1)/4.000
DO 10 I=1,NPEL
AK(I)=0.000
DO 20 K=1,2
II=12*(K-1)+1
IT=II+6
DO 20 L=II,IT,2
X1=CORRG(L,1)
Y1=CORRG(L,2)
Z1=CORRG(L,3)
W1(1,L)=DF(X,Y,Z,X1,Y1,Z1)
W1(2,L)=DF(Y,Z,X,Y1,Z1,X1)
W1(3,L)=DF(Z,X,Y,Z1,X1,Y1)
W2(L)=SFC(X,Y,Z,X1,Y1,Z1)
DO 30 K=1,2
II=(K-1)*12+2
IT=II+4
DO 30 L=II,IT,4
X1=CORRG(L,1)
Y1=CORRG(L,2)
Z1=CORRG(L,3)
W1(1,L)=D2(X,Y,Z,Y1,Z1)
W1(2,L)=D4(X,Z,Y1,Z1)

```



```

30 W1(3,L)=D4(X,Y,Z,Z1,Y1)
W2(L)=SFM(X,Y,Z,Y1,Z1)
DO 40 K=1,2
IT=(K-1)*12+4
IT=IT+4
DO 40 L=1,IT,4
X1=CORDG(L,1)
Y1=CORDG(L,2)
Z1=CORDG(L,3)
W1(1,L)=D4(Y,Z,X,X1,Z1)
W1(2,L)=D2(Y,Z,X,X1,X1)
W1(3,L)=D4(Y,X,X,Z1,X1)
W2(L)=SFM(Y,Z,X,Z1,X1)
DO 50 L=9,12
X1=CORDG(L,1)
Y1=CORDG(L,2)
Z1=CORDG(L,3)
W1(1,L)=D4(Z,Y,X1,Y1)
W1(2,L)=D4(Z,X,X1,X1)
W1(3,L)=D2(Z,X,Y,X1,Y1)
W2(L)=SFM(Z,X,Y,X1,Y1)
DO 60 I=1,3
DO 60 J=1,3
AJ(I,J)=C.GDC
AJ(I,J)=AJ(I,J)+W1(I,K)*COREL(K,J)
DO 60 K=1,NPEL
DTJ=AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2,1)+AJ(2,3)*AJ(3,2)*AJ(1,1)-AJ(3,1)*AJ(2,3)*AJ(1,1)-AJ(3,2)*AJ(2,1)*AJ(1,1)-AJ(3,3)*AJ(1,2)*AJ(2,1)
IF(N.EQ.1) GO TO 130
GO TO (70,80,90),NFACE
CRP(1)=AJ(2,2)*AJ(3,3)-AJ(2,3)*AJ(3,2)
CRP(2)=AJ(2,3)*AJ(3,1)-AJ(3,1)*AJ(2,2)
CRP(3)=AJ(2,1)*AJ(3,2)-AJ(3,2)*AJ(2,1)
GO TO 100
CRP(1)=AJ(1,3)*AJ(3,2)-AJ(1,2)*AJ(3,3)
CRP(2)=AJ(1,1)*AJ(3,3)-AJ(1,3)*AJ(3,1)
CRP(3)=AJ(1,2)*AJ(3,1)-AJ(1,1)*AJ(3,2)
GO TO 100
CRP(1)=AJ(1,2)*AJ(3,3)-AJ(1,3)*AJ(3,2)
CRP(2)=AJ(1,3)*AJ(3,1)-AJ(1,1)*AJ(3,3)
CRP(3)=AJ(1,1)*AJ(3,2)-AJ(1,2)*AJ(3,1)
CONTINUE
SIGN=+1.0DC
IF(NFACE.GT.3) SIGN=-1.0DC
DO 120 I=1,8
I1=NC(NFACE,I)
W3(I)=W2(I1)

```


GRP00950
GRP00960
GRP00970
GRP00980
GRP00990
GRP01000
GRP01010
GRP01020
GRP01030

```
120 DO 120 J=1,3  
120 AK2(I,J)=CRP(J)*W3(I)*SIGN  
130 GO TO 150  
130 CONTINUE  
140 DO 140 L=1,NPEL  
140 AK(L)=DTJ*W2(L)  
150 CONTINUE  
150 RETURN  
150 END
```


MAP00010
MAP00020
MAP00030
MAP00040
MAP00050
MAP00060
MAP00070
MAP00080
MAP00090
MAP00100
MAP00110
MAP00120
MAP00130
MAP00140
MAP00150
MAP00160
MAP00170
MAP00180
MAP00190
MAP00200
MAP00210
MAP00220
MAP00230
MAP00240
MAP00250
MAP00260
MAP00270
MAP00280
MAP00290
MAP00300
MAP00310
MAP00320
MAP00330
MAP00340
MAP00350
MAP00360
MAP00370
MAP00380
MAP00390
MAP00400
MAP00410
MAP00420
MAP00430
MAP00440
MAP00450
MAP00460

```

SUBROUTINE GRID(NX,NY,NZ,NUMNP,NPL)
IMPLICIT REAL*8(A-H,O-W)
COMMON /B2/ COORD(999,3),COREL(20,3)
DIMENSION X(100),Y(100) /
REAL*8 LABEL/8H
REAL*8 ITITLE(I2)
FORMAT(6A8)
1000 IWIDTH=9
      IXUP=15
      NY1=NY+1
      NX1=NX+1
      NY2=2*NY+1
      NX2=2*NX+1
      NY5=3*NY+2
      NZ1=NZ+1
      VXMAX=-1.0D+10
      VYMAX=-1.0D+10
      VXMIN= 1.0D+10
      VYMIN= 1.0D+10
      DO 100 I=1,NUMNP
        VXMAX=DMAX1(VXMAX,COORD(I,1))
        VYMAX=DMAX1(VYMAX,COORD(I,2))
        VXMIN=DMIN1(VXMIN,COORD(I,1))
        VYMIN=DMIN1(VYMIN,COORD(I,2))
      DO 200 I=1,NUMNP
        COORD(I,1)=COORD(I,1)-VXMIN
        COORD(I,2)=COORD(I,2)-VYMIN
      XSCALE=((VXMAX-VXMIN)/9.0)*1E+10
      YSCALE=((VYMAX-VYMIN)/15.0)*1E+10
      IF(XSCALE.LT.YSCALE) GO TO 300
      IHIGH=9
      IXUP=9
      IF(XSCALE.LE.YSCALE) XSCALE=YSCALE
      IF(XSCALE.GT.YSCALE) YSCALE=XSCALE
      IEXP=ALOG10(XSCALE)
      XSCALE=(XSCALE+(10**IEXP)*.5)*1E-10
      DO 800 LI=1,NZ1
        READ(5,1000) (ITITLE(I),I=1,6)
        READ(5,1000) (ITITLE(I),I=7,12)
        DO 600 I=1,NY1
          I1=2*I-1+(I1-1)*NPL
          DO 500 J=1,NX2/2
            J1=I1+NY5*(J-1)/2
            X(J)=COORD(J1,2)
            Y(J)=-COORD(J1,1)

```


MAP00470
 MAP00480
 MAP00490
 MAP00500
 MAP00510
 MAP00520
 MAP00530
 MAP00540
 MAP00550
 MAP00560
 MAP00570
 MAP00580
 MAP00590
 MAP00600
 MAP00610
 MAP00620
 MAP00630
 MAP00640
 MAP00650
 MAP00660
 MAP00670
 MAP00680
 MAP00690
 MAP00700
 MAP00710
 MAP00720
 MAP00730

```

    IF(J.EQ.NX2) GO TO 500
    J2=J1+NY2-I+1
    J3=J+1
    X(J3)=COORD(J2,2)
    Y(J3)=-COORD(J2,1)
500  CONTINUE
    MODCUR=2
    IF(I.EQ.1) MODCUR=1
    CALL DRAW(NX2, X, Y, MODCUR, 0, LABEL, ITITLE, XSCALE, IXUP,
    600  10,2,2, IWIDE,IHIGH,0, LAST)
    DO 800 I=1, NX1
    11=(I-1)*NY5+(L1-1)*NPL
    DO 700 J=1, NY2,2
    J1=I1+J
    X(J)=COORD(J1,2)
    Y(J)=-COORD(J1,1)
    IF(J.EQ.NY2) GO TO 700
    J2=J1+1
    J3=J+1
    X(J3)=COORD(J2,2)
    Y(J3)=-COORD(J2,1)
700  CONTINUE
    MODCUR=3
    IF(I.EQ.NX1) MODCUR=3
    800  CALL DRAW(NY2, X, Y, MODCUR, 0, LABEL, ITITLE, XSCALE, IXUP,
    10,2,2, IWIDE,IHIGH,0, LAST)
    RETURN
    END
  
```


APPENDIX D
COMPUTER LISTING (CUMEG)

CMG000430
CMG000410
CMG000430
CMG000440
CMG000450
CMG000460
CMG000470
CMG000480
CMG000490
CMG000500
CMG000510
CMG000520
CMG000530
CMG000540
CMG000550
CMG000560
CMG000570
CMG000580
CMG000590
CMG000600
CMG000610
CMG000620
CMG000630
CMG000640
CMG000650
CMG000660
CMG000670
CMG000680
CMG000690
CMG000700
CMG000710

CMG000720
CMG000730
CMG000740
CMG000750
CMG000760
CMG000770
CMG000780
CMG000790
CMG000800
CMG000810
CMG000820
CMG000830
CMG000840
CMG000850

```

2100 FORMAT(///,20X,'GRAVITY LOAD',//,14X,'NP',11X,'Z',/)
2200 FORMAT(///,22X,'PRESSURE INTENSITY =',F5.2,//27X,'EL',5X,'FACE',/)
2300 FORMAT(6X,11C,F20.12)
2400 FORMAT(11C,20X,F20.12)
2500 FORMAT(///,20X,'PRESSURE LOADS',//,14X,'NP', 9X,'X',14X,'Y',14X,
      1,7,/,/)
2600 FORMAT(110,3F20.12)
2700 FORMAT(2110)
2800 FORMAT(10A8)
2900 FORMAT(24X,15,7X,11)
      10 CONTINUE
      READ(5,1100) NX, NY, NZ, NDX, NDY, NDZ, NPDP, KORD, NCARD
      IF(NX.LT.0) STOP
      READ(5,2800) TITLE
      WRITE(6,2800) TITLE
      PI=3.141592653600/180.000
      ZRO=0.000
      ORF=1.800
      FP=4.000
      NEL = NX*NY*NZ
      NX1=NX-1
      NY1=NY-1
      NZ1=NZ-1
      NY2=NY+1
      NY3=3*NY+1
      NY4=5*NY+3
      NX2=NX+1
      NPS=NX*NY4+NY3
      NPM=2*NY2*NX2
      NPL = NPS + NPM
      NUMNP = NZ*NPL + NPS

```

COMPUTE ELEMENT CONNECTIVITY

```

DO 20 IZ=1,NZ
  IZM=IZ-1
DO 20 IX=1,NX
  IXM=IX-1
  IIR=NX-IXM
  IC=IXM*NY
  I1=IXM*NY4
  I2=I1+1
  I3=I2+1
  I4=I3+1
  I5=I1+NY3
  I6=I5+1
  I7=I6+NY
  I8=I7+1

```

C
C
C

[illegible]

CMG01340
CMG01350
CMG01360
CMG01370
CMG01380
CMG01390
CMG01400
CMG01410
CMG01420
CMG01430
CMG01440
CMG01450
CMG01460

CMG01470
CMG01480
CMG01490
CMG01500
CMG01510
CMG01520
CMG01530
CMG01540
CMG01550
CMG01560
CMG01570
CMG01580
CMG01590
CMG01600
CMG01610
CMG01620
CMG01630
CMG01640
CMG01650
CMG01660
CMG01670
CMG01680
CMG01690
CMG01700
CMG01710
CMG01720
CMG01730
CMG01740
CMG01750
CMG01760
CMG01770
CMG01780

NCON(IL,22)
NCON(IL,23)
NCON(IL,24)
NCON(IL,25)
NCON(IL,26)
NCON(IL,27)
NCON(IL,28)
NCON(IL,29)
NCON(IL,30)
NCON(IL,31)
NCON(IL,32)
NCON(IL,33)
CONTINUE

=
=
=
=
=
=
=
=
=
=
=
=
=

I27
I25
I21
I22
I23
I24
I26
I28
I32
I31
I30
1

IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL
IZM*NPL

Y
Y
Y
Y
Y
Y
Y
Y
Y
Y
Y
Y

++
++
++
++
++
++
++
++
++
++
++
++

20

COMPUTE BANDWIDTH, SOLUTION TIME, AND SPACE REQUIREMENTS

NBAND=0

NS=60

SN=NS

NDF=3

NPEL=32

NST=NPEL*NDF

NEQ=NUMNP*NDF

NN=(NEQ+NS-1)/NS

NT=(NEQ+49)/50

DO 30 I=1,NEL

DO 30 J=1,31

JP=J+1

DO 30 K=JP,NPEL

NBAND=MAX0(NBAND,IABS(NCON(I,J)-NCON(I,K)))

NBAND=(NBAND+1)*NDF

MM=(NBAND+2*(NS-1))/NS

TK=(NBAND+98)/50

MT=(1.857D0*NT**2+0.102D0*MT*NT**2-0.322D0*NT**2+4.682D0*NT*MT

1-5.23D0*MT**2)/60.0D0

FT=(50.0D0/SN)**2

KT=TK*FT+0.5D0

K7=NN*(MM+1)*96

K8=NEL*96

K11=NEQ*8

K12=NEL*32

WRITE(6,1500)

WRITE(6,1600)

DO 50 I=1,NEL

IF(NCARD.EQ.0) GO TO 40

WRITE(7,1400)(NCON(I,J), J = 1,33)

WRITE(6,1700) I,(NCON(I,J), J=1,33)

50 CONTINUE

40

50

CMG01790
CMG01800
CMG01810
CMG01820
CMG01830
CMG01840
CMG01850
CMG01860
CMG01870

CMG01880
CMG01890
CMG01900
CMG01910
CMG01920
CMG01930
CMG01940
CMG01950
CMG01960
CMG01970
CMG01980
CMG01990
CMG02000
CMG02010
CMG02020
CMG02030
CMG02040
CMG02050
CMG02060
CMG02070
CMG02080
CMG02090
CMG02100
CMG02110
CMG02120
CMG02130
CMG02140
CMG02150
CMG02160
CMG02170
CMG02180
CMG02190
CMG02200
CMG02210
CMG02220
CMG02230

```

DO 60 I=1, NUMNP
GLV(I)=ZRO
CODE(I)=ZRO
DO 60 J=1, NDF
COORD(I,J)=ZRO
PLM(I,J)=ZRO
DO 70 J=1, NPDP
READ(5,1300) I, (COORD(I,IR),IR=1,3)
70 CODE(I)=1

```

CC

CCMPUTE COORDINATES OF EDGES

```

IF (NDY.EQ.0) GO TO 90
DO 80 LA=1,NDY
READ(5,1200) I,J,N1,M1,L1
ND=3*N1
DX=(COORD(J,1)-COORD(I,1))/ND
DY=(COORD(J,2)-COORD(I,2))/ND
DZ=(COORD(J,3)-COORD(I,3))/ND
DO 80 K=1,ND
L=I+K
IF (KODE(L).GT.0) GO TO 80
KODE(L)=1
COORD(L,1)=COORD(I,1)+K*DX
COORD(L,2)=COORD(I,2)+K*DY
COORD(L,3)=COORD(I,3)+K*DZ
CONTINUE
80 IF (NDX.EQ.0) GO TO 130
DO 120 LA=1,NDX
READ(5,1200) I,J,N1,M1,L1
MD=3*M1
DX=(COORD(J,1)-COORD(I,1))/MD
DY=(COORD(J,2)-COORD(I,2))/MD
DZ=(COORD(J,3)-COORD(I,3))/MD
DO 120 K=1,M1
K1=K-1
K2=3*K-NY4
L=I+K*NY4-2
L3=L-(2*N1+1+NY)
L2=L3-NY2
IF (KODE(L2).GT.0) GO TO 100
KODE(L2)=1
COORD(L2,1)=COORD(I,1)+K2*DX
COORD(L2,2)=COORD(I,2)+K2*DY
COORD(L2,3)=COORD(I,3)+K2*DZ
IF (KODE(L3).GT.0) GO TO 110
KODE(L3)=1
100 COORD(L3,1)=COORD(L2,1)+DX

```



```

110 COORD(L3,2)=COORD(L2,2)+DY
      COORD(L3,3)=COORD(L2,3)+DZ
      IF(KODE(L).GT.0) GO TO 120
      KODE(L)=1
      COORD(L,1)=COORD(L3,1)+DX
      COORD(L,2)=COORD(L3,2)+DY
      COORD(L,3)=COORD(L3,3)+DZ
      CONTINUE
120 IF(NDZ.EQ.0) GO TO 165
130 DO 160 LA=1,NDZ
      READ(5,1200) I,J,N1,M1,L1
      LD=3*L1
      DX=(COORD(J,1)-COORD(I,1))/LD
      DY=(COORD(J,2)-COORD(I,2))/LD
      DZ=(COORD(J,3)-COORD(I,3))/LD
      DO 160 K=1,L1
      K1=K-1
      K2=3*K-2
      L=1+K*NPL
      L3=L-(M1*NY4+2*N1+1+(NX-M1)*NY2+NY)
      L2=L3-NX2*NY2
      IF(KODE(L2).GT.0) GO TO 140
      KODE(L2)=1
      COORD(L2,1)=COORD(I,1)+K2*DX
      COORD(L2,2)=COORD(I,2)+K2*DY
      COORD(L2,3)=COORD(I,3)+K2*DZ
      IF(KODE(L3).GT.0) GO TO 150
      KODE(L3)=1
      COORD(L3,1)=COORD(L2,1)+DX
      COORD(L3,2)=COORD(L2,2)+DY
      COORD(L3,3)=COORD(L2,3)+DZ
      IF(KODE(L).GT.0) GO TO 160
      KODE(L)=1
      COORD(L,1)=COORD(L3,1)+DX
      COORD(L,2)=COORD(L3,2)+DY
      COORD(L,3)=COORD(L3,3)+DZ
      CONTINUE
140
150
160
      COMPUTE COORDINATES CF CORNER NODES
      FRONT AND BACK FACES
165 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 175
      DO 170 L1=1,2
      L2=L1-1
      DO 170 ITER=1,60
      DO 170 I=1,NX1
      IC=(I-1)*NY

```

CMG02240
 CMG02250
 CMG02260
 CMG02270
 CMG02280
 CMG02290
 CMG02300
 CMG02310
 CMG02320
 CMG02340
 CMG02350
 CMG02360
 CMG02380
 CMG02390
 CMG02400
 CMG02410
 CMG02420
 CMG02430
 CMG02440
 CMG02450
 CMG02460
 CMG02470
 CMG02480
 CMG02490
 CMG02500
 CMG02510
 CMG02520
 CMG02530
 CMG02540
 CMG02550
 CMG02560
 CMG02570
 CMG02580
 CMG02590
 CMG02600
 CMG02610

CMG02620
 CMG02630
 CMG02640
 CMG02650
 CMG02660
 CMG02670

CMG02680
CMG02690
CMG02700
CMG02710
CMG02720
CMG02730
CMG02740
CMG02750
CMG02760
CMG02770
CMG02780
CMG02790
CMG02800
CMG02810
CMG02820
CMG02830
CMG02840
CMG02850
CMG02860

```

DO 170 J=1,NY1
  IL1=IC+(NEL-NX*NY)*L2
  IL2=IL1+1
  IL3=IL1+NY
  I1=1+20*L2
  I2=I1+6
  I3=I1+9
  JTI=NCON(IL1,I3)
  IF(KODE(JTI).GT.0) GO TO 170
  JTA=NCON(IL1,I1)
  JTB=NCON(IL1,I2)
  JTC=NCON(IL2,I3)
  JTD=NCON(IL3,I3)
  DO 170 N=1,3
    DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
    1-COORD(JTI,N)
    COORD(JTI,N)=COORD(JTI,N)+ORF*DX
  CONTINUE
170 CONTINUE
175 IF(NY.EQ.1).OR.(NZ.EQ.1) GO TO 185

```

C
C
C

CMG02870
CMG02880
CMG02890
CMG02900
CMG02910
CMG02920
CMG02930
CMG02940
CMG02950
CMG02960
CMG02970
CMG02980
CMG02990
CMG03000
CMG03010
CMG03020
CMG03030
CMG03040
CMG03050
CMG03060
CMG03070
CMG03080
CMG03090
CMG03100
CMG03110

```

LEFT AND RIGHT FACES

DO 180 L1=1,2
  L2=3*(L1-1)
  L3=L1-1
  DO 180 ITER=1,60
    DO 180 I=1,NZ1
      IM=I-1
      DO 180 J=1,NY1
        IL1=J+NY*(NX*(IM+L3)-L3)
        IL2=IL1+1
        IL3=IL1+NX*NY
        I1=24-L2
        I2=7+L2
        I3=27+L2
        JTI=NCON(IL1,I3)
        IF(KODE(JTI).GT.0) GO TO 180
        JTA=NCON(IL1,I1)
        JTB=NCON(IL1,I2)
        JTC=NCON(IL2,I3)
        JTD=NCON(IL3,I3)
        DO 180 N=1,3
          DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
          1-COORD(JTI,N)
          COORD(JTI,N)=COORD(JTI,N)+ORF*DX
        CONTINUE
180 CONTINUE
185 IF(NX.EQ.1).OR.(NZ.EQ.1) GO TO 195

```

C

C

TOP AND BOTTOM FACES

```

DO 190 L1=1,2
L2=L1-1
DO 190 I=1,60
DO 190 I=1,NZ1
IC=(I-1)*NX*NY+L2*NY1+1
DO 190 J=1,NX1
IL1=IC+NY*(J-1)
IL2=IL1+NX*NY
IL3=IL1+NY
I1=1+9*L2
I2=24+3*L2
I3=I1+2G
JTI=NCON(IL1,I3)
IF(KODE(JTI).GT.0) GO TO 190
JTA=NCON(IL1,I2)
JTB=NCON(IL1,I1)
JTC=NCON(IL2,I3)
JTD=NCON(IL3,I3)
DO 190 N=1,3
DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
1-COORD(JTI,N)
COORD(JTI,N)=COORD(JTI,N)+ORF*DX
CONTINUE
190 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 205
195

```

C

INTERIOR FACES

```

DO 200 I=1,NZ1
DO 200 I=1,60
DO 200 J=1,NX1
IC=(J-1)*NY+(I-1)*NX*NY
DO 200 K=1,NY1
IL1=IC+K
IL2=IL1+1
IL3=IL1+NY
JTI=NCON(IL1,30)
IF(KODE(JTI).GT.0) GO TO 200
JTA=NCON(IL1,21)
JTB=NCON(IL1,27)
JTC=NCON(IL2,30)
JTD=NCON(IL3,30)
DO 200 N=1,3
DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP
1-COORD(JTI,N)
COORD(JTI,N)=COORD(JTI,N)+ORF*DX
CONTINUE
200

```

C

CMG03120
CMG03130
CMG03140
CMG03150
CMG03160
CMG03170
CMG03180
CMG03190
CMG03200
CMG03210
CMG03220
CMG03230
CMG03240
CMG03250
CMG03260
CMG03270
CMG03280
CMG03290
CMG03300
CMG03310
CMG03320
CMG03330
CMG03340
CMG03350

CMG03360
CMG03370
CMG03380
CMG03390
CMG03400
CMG03410
CMG03420
CMG03430
CMG03440
CMG03450
CMG03460
CMG03470
CMG03480
CMG03490
CMG03500
CMG03510
CMG03520
CMG03530
CMG03540


```

C
C
C
C      COMPUTE COORDINATES OF MID-SIDE NODES
205  DO 230 L=1,NEL
      K=12
      DO 210 I=1,10,3
        I1=I+20
        I2=I+K
        I3=I2+4
        K=K-2
        JT1=NCON(L,I2)
        IF(KODE(JT1).GT.0) GO TO 210
        JT2=NCON(L,I3)
        JT3=NCON(L,I1)
        JT4=NCON(L,I)
        DO 210 N=1,3
          DX=(COORD(JT3,N)-COORD(JT4,N))/3.000
          COORD(JT1,N)=COORD(JT4,N)+DX
          COORD(JT2,N)=COORD(JT1,N)+DX
210  CONTINUE
      DO 220 I=1,2
        DO 220 J=1,7,6
          I1=J+(I-1)*20
          I2=I1+1
          I3=I2+1
          I4=I3+1
          JT1=NCON(L,I2)
          IF(KODE(JT1).GT.0) GO TO 220
          JT2=NCON(L,I3)
          JT3=NCON(L,I4)
          JT4=NCON(L,I1)
          DO 220 N=1,3
            DX=(COORD(JT3,N)-COORD(JT4,N))/3.000
            COORD(JT1,N)=COORD(JT4,N)+DX
            COORD(JT2,N)=COORD(JT1,N)+DX
220  CONTINUE
      DO 230 I=1,2
        K=9
        K1=1
        DO 230 J=1,4,3
          I1=J+(I-1)*20
          I2=I1+K
          I3=I2+K1
          I4=I3+K1
          K=3
          K1=-1
          JT1=NCON(L,I3)
          IF(KODE(JT1).GT.0) GO TO 230

```

```

CMG03550
CMG03560
CMG03570
CMG03580
CMG03590
CMG03600
CMG03610
CMG03620
CMG03630
CMG03640
CMG03650
CMG03660
CMG03670
CMG03680
CMG03690
CMG03700
CMG03710
CMG03720
CMG03730
CMG03740
CMG03750
CMG03760
CMG03770
CMG03780
CMG03790
CMG03800
CMG03810
CMG03820
CMG03830
CMG03840
CMG03850
CMG03860
CMG03870
CMG03880
CMG03890
CMG03900
CMG03910
CMG03920
CMG03930
CMG03940
CMG03950
CMG03960
CMG03970
CMG03980
CMG03990

```


CMG04090
CMG00410
CMG04020
CMG00430
CMG04050
CMG04060
CMG04070
CMG04080
CMG04090
CMG04100
CMG04105
CMG04110
CMG04120
CMG04130
CMG04140
CMG04150
CMG04160
CMG04170
CMG04180
CMG04190
CMG04200
CMG04210

CMG04220
CMG04230
CMG04240
CMG04250
CMG04260
CMG04290
CMG04300
CMG04310
CMG04320
CMG04330
CMG04340
CMG04350
CMG04360
CMG04370
CMG04380
CMG04390
CMG04400

CMG04410
CMG04420
CMG04430

```

230 JT2=NCON(L,I4)
    JT3=NCON(L,I1)
    JT4=NCON(L,I2)
    DO 230 N=1,3
    DX=(COORD(JT3,N)-COORD(JT4,N))/3.0D0
    COORD(JT1,N)=COORD(JT4,N)+DX
    COORD(JT2,N)=COORD(JT1,N)+DX
    CONTINUE
    WRITE(6,200C)
    DO 250 I=1,NUMNP
    KODE(I) = 0
    IF(KORD.EQ.0) GO TO 240
    PHI=PI*COORD(I,2)
    RAD=COORD(I,1)
    COORD(I,1)=RAD*DCOS(PHI)
    COORD(I,2)=RAD*DSIN(PHI)
    IF(NCARD.EQ.0) GO TO 250
    WRITE(7,180C) I,(COORD(I,L),L=1,3)
    240 WRITE(6,180C) I,(COORD(I,L),L=1,3)
    250 READ(5,190C) MAP,NEFP,SGZ,UDP
    IF((SGZ.EQ.0.0).AND.(NEFP.EQ.0.0)) GO TO 340
    IF(SGZ.EQ.0.0) GO TO 290

```

DETERMINE CONSISTENT GRAVITY LOAD

```

DO 270 I=1,NEL
DO 260 L=1,NPEL
J2 = NCON(I,L)
DO 260 K=1,NDF
    COREL(L,K) = COORD(J2,K)
    260 CALL CUB5V
    DO 270 J=1,NPEL
    J1=NCON(I,J)
    KODE(J1) = 1
    GLV(J1)=GLV(J1)+SGZ*AK1(J)
    270 CONTINUE
    WRITE(6,210C)
    DO 280 I=1,NUMNP
    IF((NCARD.EQ.0).OR.(NEFP.NE.0)) GO TO 280
    WRITE(7,240C) I,GLV(I)
    280 WRITE(6,230C) I,GLV(I)
    290 IF(NEFP.EQ.0) GO TO 340

```

DETERMINE CONSISTENT PRESSURE LOAD

```

WRITE(6,220C) UDP
DO 310 I=1,NEFP
READ(5,270C) NELP,NFACE

```



```

WRITE(6,2900) NELP,NFACE
DO 300 J=1,12
  J1=NC(NFACE,J)
  NELC(J)=NCON(NELP,J1)
  DO 305 L=1,NPEL
    J2 = NCON(NELP,L)
    DO 305 K=1,NDF
      COREL(L,K) = COORD(J2,K)
    CALL QUAD5
  DO 310 J=1,12
    J1=NELC(J)
    KCODE(J1) = 1
    DO 310 L=1,NDF
      PLM(J1,L)=PLM(J1,L)+UDP*AK3(J,L)
    WRITE(6,2500)
  DO 330 I=1,NUMNP
    IF(KCODE(I).EQ.0) GO TO 330
    IF(NCARD.EQ.0) GO TO 320
    PLM(I,3)=PLM(I,3)+GLV(I)
    WRITE(7,2600) I,(PLM(I,J),J=1,3)
    WRITE(6,1800) I,(PLM(I,L),L=1,3)
  CONTINUE
CONTINUE
WRITE(6,1000)
IF(MAP.EQ.0) GO TO 350
C
C
C
DRAW STRUCTURE MESH
CALL GRID(NX,NY,NZ,NUMNP,NPL)
CONTINUE
GO TO 10
END

```

CMG04440
 CMG04450
 CMG04460
 CMG04470
 CMG04471
 CMG04472
 CMG04473
 CMG04474
 CMG04480
 CMG04490
 CMG04500
 CMG04510
 CMG04520
 CMG04530
 CMG04540
 CMG04550
 CMG04560
 CMG04570
 CMG04580
 CMG04590
 CMG04600
 CMG04610
 CMG04620
 CMG04630
 CMG04640

CMG04650
 CMG04660
 CMG04670
 CMG04680


```

SUBROUTINE CUB5V
IMPLICIT REAL*8(A-H,O-Z)
COMMON /NB1/ NPTEL,NEL,NFACE,NUMNP
COMMON /B3/ AK(32),AK1(32)
DIMENSION XI(5),AI(5),AIA(5,5,5)
DATA XI/0.9061798459386640,0.5384693101056831,0.0D0,-0.53846931010
156831,-0.9061798459386640/
DATA AI/0.2369268850561891,0.4786286704993665,0.5688888888888889,0
1.4786286704993665,0.2369268850561891/
DO 100 I=1,NPTEL
100 AK1(I)=0.0D0
DO 200 I=1,5
DO 200 J=1,5
DO 200 K=1,5
200 AIA(I,J,K)=AI(I)*AI(J)*AI(K)
X=XI(I)
DO 400 J=1,5
Y=XI(J)
DO 400 K=1,5
Z=XI(K)
CALL GRAP(X,Y,Z,1)
DO 300 L=1,NPTEL
300 AK1(L)=AK1(L)+AIA(I,J,K)*AK(L)
400 CONTINUE
RETURN
END
CUB000010
CUB000020
CUB000030
CUB000040
CUB000050
CUB000060
CUB000070
CUB000080
CUB000090
CUB000100
CUB000110
CUB000120
CUB000130
CUB000140
CUB000150
CUB000160
CUB000170
CUB000180
CUB000190
CUB000200
CUB000210
CUB000220
CUB000230
CUB000240
CUB000250
CUB000260
CUB000270

```



```

SUBROUTINE GRAP(X,Y,Z,N)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /NB1/ NPEL,NEL,NFACE,NUMNP
COMMON /B2/ COORD(1050,3),COREL(32,3)
COMMON /B3/ AK(32),AK1(32)
COMMON /B4/ AK2(12,3),AK3(12,3)
DIMENSION W1(3,32),W2(32),W3(12),AJ(3,3),CORDG(32,3),
1CRP(3),NC(6,12)
1DATA CORDG(1,1),CORDG(1,2),CORDG(1,3),CORDG(1,4),CORDG(1,5),CORDG(1,6),CORDG(1,7),CORDG(1,8),CORDG(1,9),CORDG(1,10),CORDG(1,11),CORDG(1,12),CORDG(1,13),CORDG(1,14),CORDG(1,15),CORDG(1,16),CORDG(1,17),CORDG(1,18),CORDG(1,19),CORDG(1,20),CORDG(1,21),CORDG(1,22),CORDG(1,23),CORDG(1,24),CORDG(1,25),CORDG(1,26),CORDG(1,27),CORDG(1,28),CORDG(1,29),CORDG(1,30),CORDG(1,31),CORDG(1,32),CORDG(1,33),CORDG(1,34),CORDG(1,35),CORDG(1,36),CORDG(1,37),CORDG(1,38),CORDG(1,39),CORDG(1,40),CORDG(1,41),CORDG(1,42),CORDG(1,43),CORDG(1,44),CORDG(1,45),CORDG(1,46),CORDG(1,47),CORDG(1,48),CORDG(1,49),CORDG(1,50),CORDG(1,51),CORDG(1,52),CORDG(1,53),CORDG(1,54),CORDG(1,55),CORDG(1,56),CORDG(1,57),CORDG(1,58),CORDG(1,59),CORDG(1,60),CORDG(1,61),CORDG(1,62),CORDG(1,63),CORDG(1,64),CORDG(1,65),CORDG(1,66),CORDG(1,67),CORDG(1,68),CORDG(1,69),CORDG(1,70),CORDG(1,71),CORDG(1,72),CORDG(1,73),CORDG(1,74),CORDG(1,75),CORDG(1,76),CORDG(1,77),CORDG(1,78),CORDG(1,79),CORDG(1,80),CORDG(1,81),CORDG(1,82),CORDG(1,83),CORDG(1,84),CORDG(1,85),CORDG(1,86),CORDG(1,87),CORDG(1,88),CORDG(1,89),CORDG(1,90),CORDG(1,91),CORDG(1,92),CORDG(1,93),CORDG(1,94),CORDG(1,95),CORDG(1,96),CORDG(1,97),CORDG(1,98),CORDG(1,99),CORDG(1,100),CORDG(1,101),CORDG(1,102),CORDG(1,103),CORDG(1,104),CORDG(1,105),CORDG(1,106),CORDG(1,107),CORDG(1,108),CORDG(1,109),CORDG(1,110),CORDG(1,111),CORDG(1,112),CORDG(1,113),CORDG(1,114),CORDG(1,115),CORDG(1,116),CORDG(1,117),CORDG(1,118),CORDG(1,119),CORDG(1,120),CORDG(1,121),CORDG(1,122),CORDG(1,123),CORDG(1,124),CORDG(1,125),CORDG(1,126),CORDG(1,127),CORDG(1,128),CORDG(1,129),CORDG(1,130),CORDG(1,131),CORDG(1,132),CORDG(1,133),CORDG(1,134),CORDG(1,135),CORDG(1,136),CORDG(1,137),CORDG(1,138),CORDG(1,139),CORDG(1,140),CORDG(1,141),CORDG(1,142),CORDG(1,143),CORDG(1,144),CORDG(1,145),CORDG(1,146),CORDG(1,147),CORDG(1,148),CORDG(1,149),CORDG(1,150),CORDG(1,151),CORDG(1,152),CORDG(1,153),CORDG(1,154),CORDG(1,155),CORDG(1,156),CORDG(1,157),CORDG(1,158),CORDG(1,159),CORDG(1,160),CORDG(1,161),CORDG(1,162),CORDG(1,163),CORDG(1,164),CORDG(1,165),CORDG(1,166),CORDG(1,167),CORDG(1,168),CORDG(1,169),CORDG(1,170),CORDG(1,171),CORDG(1,172),CORDG(1,173),CORDG(1,174),CORDG(1,175),CORDG(1,176),CORDG(1,177),CORDG(1,178),CORDG(1,179),CORDG(1,180),CORDG(1,181),CORDG(1,182),CORDG(1,183),CORDG(1,184),CORDG(1,185),CORDG(1,186),CORDG(1,187),CORDG(1,188),CORDG(1,189),CORDG(1,190),CORDG(1,191),CORDG(1,192),CORDG(1,193),CORDG(1,194),CORDG(1,195),CORDG(1,196),CORDG(1,197),CORDG(1,198),CORDG(1,199),CORDG(1,200),CORDG(1,201),CORDG(1,202),CORDG(1,203),CORDG(1,204),CORDG(1,205),CORDG(1,206),CORDG(1,207),CORDG(1,208),CORDG(1,209),CORDG(1,210),CORDG(1,211),CORDG(1,212),CORDG(1,213),CORDG(1,214),CORDG(1,215),CORDG(1,216),CORDG(1,217),CORDG(1,218),CORDG(1,219),CORDG(1,220),CORDG(1,221),CORDG(1,222),CORDG(1,223),CORDG(1,224),CORDG(1,225),CORDG(1,226),CORDG(1,227),CORDG(1,228),CORDG(1,229),CORDG(1,230),CORDG(1,231),CORDG(1,232),CORDG(1,233),CORDG(1,234),CORDG(1,235),CORDG(1,236),CORDG(1,237),CORDG(1,238),CORDG(1,239),CORDG(1,240),CORDG(1,241),CORDG(1,242),CORDG(1,243),CORDG(1,244),CORDG(1,245),CORDG(1,246),CORDG(1,247),CORDG(1,248),CORDG(1,249),CORDG(1,250),CORDG(1,251),CORDG(1,252),CORDG(1,253),CORDG(1,254),CORDG(1,255),CORDG(1,256),CORDG(1,257),CORDG(1,258),CORDG(1,259),CORDG(1,260),CORDG(1,261),CORDG(1,262),CORDG(1,263),CORDG(1,264),CORDG(1,265),CORDG(1,266),CORDG(1,267),CORDG(1,268),CORDG(1,269),CORDG(1,270),CORDG(1,271),CORDG(1,272),CORDG(1,273),CORDG(1,274),CORDG(1,275),CORDG(1,276),CORDG(1,277),CORDG(1,278),CORDG(1,279),CORDG(1,280),CORDG(1,281),CORDG(1,282),CORDG(1,283),CORDG(1,284),CORDG(1,285),CORDG(1,286),CORDG(1,287),CORDG(1,288),CORDG(1,289),CORDG(1,290),CORDG(1,291),CORDG(1,292),CORDG(1,293),CORDG(1,294),CORDG(1,295),CORDG(1,296),CORDG(1,297),CORDG(1,298),CORDG(1,299),CORDG(1,300),CORDG(1,301),CORDG(1,302),CORDG(1,303),CORDG(1,304),CORDG(1,305),CORDG(1,306),CORDG(1,307),CORDG(1,308),CORDG(1,309),CORDG(1,310),CORDG(1,311),CORDG(1,312),CORDG(1,313),CORDG(1,314),CORDG(1,315),CORDG(1,316),CORDG(1,317),CORDG(1,318),CORDG(1,319),CORDG(1,320),CORDG(1,321),CORDG(1,322),CORDG(1,323),CORDG(1,324),CORDG(1,325),CORDG(1,326),CORDG(1,327),CORDG(1,328),CORDG(1,329),CORDG(1,330),CORDG(1,331),CORDG(1,332),CORDG(1,333),CORDG(1,334),CORDG(1,335),CORDG(1,336),CORDG(1,337),CORDG(1,338),CORDG(1,339),CORDG(1,340),CORDG(1,341),CORDG(1,342),CORDG(1,343),CORDG(1,344),CORDG(1,345),CORDG(1,346),CORDG(1,347),CORDG(1,348),CORDG(1,349),CORDG(1,350),CORDG(1,351),CORDG(1,352),CORDG(1,353),CORDG(1,354),CORDG(1,355),CORDG(1,356),CORDG(1,357),CORDG(1,358),CORDG(1,359),CORDG(1,360),CORDG(1,361),CORDG(1,362),CORDG(1,363),CORDG(1,364),CORDG(1,365),CORDG(1,366),CORDG(1,367),CORDG(1,368),CORDG(1,369),CORDG(1,370),CORDG(1,371),CORDG(1,372),CORDG(1,373),CORDG(1,374),CORDG(1,375),CORDG(1,376),CORDG(1,377),CORDG(1,378),CORDG(1,379),CORDG(1,380),CORDG(1,381),CORDG(1,382),CORDG(1,383),CORDG(1,384),CORDG(1,385),CORDG(1,386),CORDG(1,387),CORDG(1,388),CORDG(1,389),CORDG(1,390),CORDG(1,391),CORDG(1,392),CORDG(1,393),CORDG(1,394),CORDG(1,395),CORDG(1,396),CORDG(1,397),CORDG(1,398),CORDG(1,399),CORDG(1,400),CORDG(1,401),CORDG(1,402),CORDG(1,403),CORDG(1,404),CORDG(1,405),CORDG(1,406),CORDG(1,407),CORDG(1,408),CORDG(1,409),CORDG(1,410),CORDG(1,411),CORDG(1,412),CORDG(1,413),CORDG(1,414),CORDG(1,415),CORDG(1,416),CORDG(1,417),CORDG(1,418),CORDG(1,419),CORDG(1,420),CORDG(1,421),CORDG(1,422),CORDG(1,423),CORDG(1,424),CORDG(1,425),CORDG(1,426),CORDG(1,427),CORDG(1,428),CORDG(1,429),CORDG(1,430),CORDG(1,431),CORDG(1,432),CORDG(1,433),CORDG(1,434),CORDG(1,435),CORDG(1,436),CORDG(1,437),CORDG(1,438),CORDG(1,439),CORDG(1,440),CORDG(1,441),CORDG(1,442),CORDG(1,443),CORDG(1,444),CORDG(1,445),CORDG(1,446),CORDG(1,447),CORDG(1,448),CORDG(1,449),CORDG(1,
```



```

20  Z1=CORDG(J,3)
    W2(J)=SFC(X,Y,Z,X1,Y1,Z1)
    W1(1,J)=DFX(X,Y,Z,X1,Y1,Z1)
    W1(2,J)=DFX(Y,X,Z,Y1,X1,Z1)
    W1(3,J)=DFX(Z,X,Y,Z1,X1,Y1)
    DO 30 K=1,2
    IK=20*(K-1)
    DO 30 I=1,2
    II=6*(I-1)+2+IK
    IT=II+1
    DO 30 J=1,IT
    X1=CORDG(J,1)
    Y1=CORDG(J,2)
    Z1=CORDG(J,3)
    W2(J)=SFM(X,Y,Z,X1,Y1,Z1)
    W1(1,J)=DX(X,Y,Z,X1,Y1,Z1)
    W1(2,J)=DY(X,Z,X1,Y1,Z1)
    W1(3,J)=DZ(X,Y,X1,Y1,Z1)
    DO 40 K=1,2
    IK=20*(K-1)
    DO 40 I=1,2
    II=6*(I-1)+5+IK
    IT=II+1
    DO 40 J=1,IT
    X1=CORDG(J,1)
    Y1=CORDG(J,2)
    Z1=CORDG(J,3)
    W2(J)=SFM(Y,Z,X,Y1,Z1,X1)
    W1(1,J)=DY(Y,Z,Y1,X1,Z1)
    W1(2,J)=DX(Y,X,Z,Y1,X1,Z1)
    W1(3,J)=DZ(Y,X,Y1,X1,Z1)
    DO 50 K=1,2
    IK=4*(K-1)
    II=13+IK
    IT=II+3
    DO 50 J=1,IT
    X1=CORDG(J,1)
    Y1=CORDG(J,2)
    Z1=CORDG(J,3)
    W2(J)=SFM(Z,X,Y,Z1,X1,Y1)
    W1(1,J)=DZ(Z,Y,X,Z1,Y1,X1)
    W1(2,J)=DY(Z,X,Z1,Y1,X1)
    W1(3,J)=DX(Z,Y,X,Z1,Y1,X1)
    DO 60 I=1,3
    DO 60 J=1,3
    AJ(I,J)=Q.000
    DO 60 K=1,NPEL
    60  AJ(I,J)=AJ(I,J)+W1(I,K)*COREL(K,J)

```

```

GRP00470
GRP00480
GRP00490
GRP00500
GRP00510
GRP00520
GRP00530
GRP00540
GRP00550
GRP00560
GRP00570
GRP00580
GRP00590
GRP00600
GRP00610
GRP00620
GRP00630
GRP00640
GRP00650
GRP00660
GRP00670
GRP00680
GRP00690
GRP00700
GRP00710
GRP00720
GRP00730
GRP00740
GRP00750
GRP00760
GRP00770
GRP00780
GRP00790
GRP00800
GRP00810
GRP00820
GRP00830
GRP00840
GRP00850
GRP00860
GRP00870
GRP00880
GRP00890
GRP00900
GRP00910
GRP00920
GRP00930
GRP00940

```



```

DTJ=AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,1)*AJ(2,3)*AJ(3,3)*
1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(2,3)*AJ(1,1)-AJ(3,1)*AJ(2,3)*AJ(3,3)*
2AJ(2,1)*AJ(1,2)
IF(N.EQ.1) GO TO 130
GO TO (70,80,90,100,110,120,130),NFAC
CRP(1)=AJ(2,2)*AJ(3,3)-AJ(2,3)*AJ(3,2)
CRP(2)=AJ(2,3)*AJ(3,1)-AJ(2,1)*AJ(3,3)
CRP(3)=AJ(2,1)*AJ(3,2)-AJ(2,2)*AJ(3,1)
GO TO 100
CRP(1)=AJ(1,3)*AJ(3,2)-AJ(1,2)*AJ(3,3)
CRP(2)=AJ(1,1)*AJ(3,3)-AJ(1,3)*AJ(3,1)
CRP(3)=AJ(1,2)*AJ(3,1)-AJ(1,1)*AJ(3,2)
GO TO 100
CRP(1)=AJ(1,2)*AJ(2,3)-AJ(1,3)*AJ(2,2)
CRP(2)=AJ(1,3)*AJ(2,1)-AJ(1,1)*AJ(2,3)
CRP(3)=AJ(1,1)*AJ(2,2)-AJ(1,2)*AJ(2,1)
CONTINUE
SIGN=+1.0D0
IF(NFAC.EGT.3) SIGN=-1.0D0
DO 120 I=1,12
I1=NC(NFACE,I)
W3(I)=W2(I1)
DC 120 J=1,3
AK2(I,J)=CRP(J)*W3(I)*SIGN
120 GO TO 150
CONTINUE
DO 140 L=1,NPEL
AK(L)=DTJ*W2(L)
140 CONTINUE
150 RETURN
END
GRPO0950
GRPO0960
GRPO0970
GRPO0980
GRPO0990
GRPO1000
GRPO1010
GRPO1020
GRPO1030
GRPO1040
GRPO1050
GRPO1060
GRPO1070
GRPO1080
GRPO1090
GRPO1100
GRPO1110
GRPO1120
GRPO1130
GRPO1140
GRPO1150
GRPO1160
GRPO1170
GRPO1180
GRPO1190
GRPO1200
GRPO1210
GRPO1220
GRPO1230
GRPO1240
GRPO1250

```



```

SUBROUTINE GRID(NX,NY,NZ,NUMNP,NPL)
IMPLICIT REAL*8(A-H,O-W)
COMMON /B2/ COORD(1050,3),COREL(32,3)
DIMENSION X(100),Y(100) /
REAL*8 LABEL/8H
REAL*8 ITITLE(12)
FORMAT(6A8)
1000 IWIDTH=9
IHIGH=15
IXUP=15
NY1=NY+1
NX1=NX+1
NZ1=NZ+1
NY2=3*NY+1
NX2=3*NX+1
NY3=5*NY+3
VXMAX=-1.0D+10
VYMAX=-1.0D+10
VXMIN=1.0D+10
VYMIN=1.0D+10
DO 100 I=1,NUMNP
VXMAX=DMAX1(VXMAX,COORD(I,1))
VYMAX=DMAX1(VYMAX,COORD(I,2))
VXMIN=DMIN1(VXMIN,COORD(I,1))
VYMIN=DMIN1(VYMIN,COORD(I,2))
DO 200 I=1,NUMNP
COORD(I,1)=COORD(I,1)-VXMIN
COORD(I,2)=COORD(I,2)-VYMIN
XSCALE=((VXMAX-VXMIN)/9.0)*1.0E+10
YSCALE=((VYMAX-VYMIN)/15.0)*1.0E+10
IF(XSCALE.LT.YSCALE) GO TO 300
IHIGH=9
IXUP=9
IF(XSCALE.LE.YSCALE) XSCALE=YSCALE
IF(XSCALE.GT.YSCALE) YSCALE=XSCALE
IEXP=ALOG10(XSCALE)
XSCALE=(XSCALE+(10*IEXP)*.5)*1.0E-10
DO 700 L1=1,NZ1
READ(5,1000) (ITITLE(I),I=1,6)
READ(5,1000) (ITITLE(I),I=7,12)
DO 500 I=1,NY1
I1=3*I-2+(L1-1)*NPL
DO 400 J=1,NX2,3
J1=I1+NY3*(J-1)/3
X(J)=COORD(J1,2)
Y(J)=-COORD(J1,1)

```

MAP00010
MAP00020
MAP00030
MAP00040
MAP00050
MAP00060
MAP00070
MAP00080
MAP00090
MAP00100
MAP00110
MAP00120
MAP00130
MAP00140
MAP00150
MAP00160
MAP00170
MAP00180
MAP00190
MAP00200
MAP00210
MAP00220
MAP00230
MAP00240
MAP00250
MAP00260
MAP00270
MAP00280
MAP00290
MAP00300
MAP00310
MAP00320
MAP00330
MAP00340
MAP00350
MAP00360
MAP00370
MAP00380
MAP00390
MAP00400
MAP00410
MAP00420
MAP00430
MAP00440
MAP00450
MAP00460

MAP00470
 MAP00480
 MAP00490
 MAP00500
 MAP00510
 MAP00520
 MAP00530
 MAP00540
 MAP00550
 MAP00560
 MAP00570
 MAP00580
 MAP00590
 MAP00600
 MAP00610
 MAP00620
 MAP00630
 MAP00640
 MAP00650
 MAP00660
 MAP00670
 MAP00680
 MAP00690
 MAP00700
 MAP00710
 MAP00720
 MAP00730
 MAP00740
 MAP00750
 MAP00760
 MAP00770
 MAP00780
 MAP00790
 MAP00800
 MAP00810

```

IF(J.EQ.NX2) GO TO 400
J2=J1+NY2-2*(I-1)
J3=J2+NY1
J4=J+1
J5=J4+1
X(J4)=COORD(J2,2)
Y(J4)=-COORD(J2,1)
X(J5)=COORD(J3,2)
Y(J5)=-COORD(J3,1)
400 CONTINUE
MODCUR=2
IF(I.EQ.1) MODCUR=1
500 CALL DRAW(NX2, X, Y, MODCUR, 0, LABEL, ITITLE, XSCALE, IXUP,
10,2,2, IWIDTH, IHIGH, 0, LAST)
DO 700 I=1, NX1
I1=(I-1)*NY3+(I1-1)*NPL
DO 600 J=1, NY2, 3
J1=I1+J
X(J1)=COORD(J1,2)
Y(J1)=-COORD(J1,1)
IF(J.EQ.NY2) GO TO 600
J2=J1+1
J3=J2+1
J4=J+1
J5=J4+1
X(J4)=COORD(J2,2)
Y(J4)=-COORD(J2,1)
X(J5)=COORD(J3,2)
Y(J5)=-COORD(J3,1)
600 CONTINUE
IF(I.EQ.NX1) MODCUR=3
700 CALL DRAW(NY2, X, Y, MODCUR, 0, LABEL, ITITLE, XSCALE, IXUP,
10,2,2, IWIDTH, IHIGH, 0, LAST)
RETURN
END

```


REFERENCES

1. Akyuz, F.A., "Natural Coordinate Systems: An Automatic Input Data Generation Scheme for a Finite-Element Method," Nuclear Engineering and Design, Vol. 11, No. 2, pp 195 - 207, March 1970. '69
2. Becker, E.B. and Brisband, J.J., Rohm and Haas Company, Huntsville, Alabama, Technical Report S-76, Application of the Finite Element Method to Stress Analysis of Solid Propellant Rocket Grains, Vol. 1, November 18, 1965, Vol. 2, pt 1, January 21, 1966.
3. Cantin, G., "Rigid Body Motions in Curved Finite Elements," AIAA Journal, Vol. 8, No. 7, pp 1252 - 1255, July, 1970.
4. Cantin, G., "Three-Dimensional Finite Element Studies," (Naval Postgraduate School report in preparation).
5. Dailey, G., The Johns Hopkins University Technical Memorandum TG 1128, A Computer Program to Plot Contour Maps of Stresses in Two-Dimensional Structures Analyzed by the Method of Finite Elements, July, 1970.
6. Dailey, G., The Johns Hopkins University Technical Memorandum TG 1140, Computer Programs for Plotting Finite-Element Patterns of Two- and Three-Dimensional Structures, October, 1970.
7. Holand, I. and Bell, K., Finite Element Methods in Stress Analysis, Chapter 13 by Zienkiewicz, Irons, Ergatoudis, Ahmad, and Scott, pp 383 - 435, Tapir, Technical University of Norway, Trondheim, Norway, 1969.
8. Jones, R.L., National Aeronautics and Space Administration Technical Note D - 6022, A Generalized Digital Contouring Program, January, 1971.
9. Sprague, R. L., National Aeronautics and Space Administration Technical Memorandum X - 2204, GIG - A Fortran IV Geometry Input Generator Program for a Finite-Element Heat-Transfer Code, March, 1971.
10. Serpanos, J.E., Naval Weapons Center, China Lake, California Technical Note 4062 - 47, SAGPAC - Structural Analysis Graphics Package, A Fortran IV Computer Program, June, 1970.
11. Stroud, A.H. and Secrest, D., Gaussian Quadrature Formulas, Prentice-Hall Inc., 1966.

12. Users Manual, W.R. Church Computer Center, U.S. Naval Postgraduate School, Monterey, California, 1970.
13. Zienkiewicz, O.C. and Cheung, Y.K., The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company, Ltd., London, 1967.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Professor Gilles Cantin, Code 59 Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
4. Professor Robert E. Newton, Code 59 Ne Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
5. Assistant Professor David Salinas, Code 59 Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6. LT Dale E. Hanson, USN U.S. Naval Ship Repair Facility Box 34 FPO San Francisco 96651	1
7. LT Emmanuel Leonidas SMC 1335 Naval Postgraduate School Monterey, California 93940	1
8. LT Charles G. Pfeifer, USN SMC 1346 U.S. Naval Postgraduate School Monterey, California 93940	1
9. Mr. D. Talapatre Department of Mechanical Engineering University of British Columbia Vancouver 8, Canada	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE Three-Dimensional Elasto-Static Problems Using Isoparametric Finite Elements			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates) Master's Thesis; June 1971			
5. AUTHOR(S) (First name, middle initial, last name) Dale E. Hanson			
6. REPORT DATE June 1971		7a. TOTAL NO. OF PAGES 93	7b. NO. OF REFS 13
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT			

The objective of the project described in this report was to develop a computer system which would generate the required input data for a structural analysis of three-dimensional elasto-static problems using isoparametric finite elements. Element connectivity, nodal point coordinates, consistent gravity loads, and consistent pressure loads are generated. A variety of algorithms are used in the system to reduce the amount of input data required. The computer system and a sample problem are discussed.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Finite Element Mesh Generator						

27 MAR 72
23 JUN 72
16 APR 73
5 AUG 80

19537
20468
20901
27087

Thesis
H2043 Hanson
c.1

128066

Three dimensional
elasto-static prob-
lems using isopara-
metric finite ele-
ments.

27 MAR 72
23 JUN 72
16 APR 73
5 AUG 80

19537
20468
20901
27087

Thesis
H2043 Hanson
c.1

128066

Three dimensional
elasto-static prob-
lems using isopara-
metric finite ele-
ments.

thesH2043

Three dimensional elasto-static problems



3 2768 002 07652 3
DUDLEY KNOX LIBRARY